



Estimation de paramètres évoluant sur des groupes de Lie : application à la cartographie et à la localisation d'une caméra monoculaire

Guillaume Bourmaud

► To cite this version:

Guillaume Bourmaud. Estimation de paramètres évoluant sur des groupes de Lie : application à la cartographie et à la localisation d'une caméra monoculaire. Instrumentations et Détecteurs [physics.ins-det]. Université de Bordeaux, 2015. Français. NNT : 2015BORD0242 . tel-01271029

HAL Id: tel-01271029

<https://theses.hal.science/tel-01271029>

Submitted on 8 Feb 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE PRÉSENTÉE
POUR OBTENIR LE GRADE DE
DOCTEUR DE
L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DES SCIENCES PHYSIQUES ET DE L'INGÉNIEUR
SPÉCIALITÉ AUTOMATIQUE, PRODUCTIQUE, SIGNAL ET IMAGE

Par Guillaume BOURMAUD

**ESTIMATION DE PARAMÈTRES ÉVOLUANT SUR DES
GROUPES DE LIE : APPLICATION À LA CARTOGRAPHIE
ET À LA LOCALISATION D'UNE CAMÉRA MONOCULAIRE**

Sous la direction de : Yannick BERTHOUMIEU

Soutenue le 6 novembre 2015

Membres du jury :

M. PEREZ, Patrick
M. BONNABEL, Sylvère
M. MONASSE, Pascal
M. BERTHOUMIEU, Yannick
Mme GIREMUS, Audrey
M. MEGRET, Rémi

Directeur de Recherche INRIA, Technicolor
Maître de Conférences HDR, Mines ParisTech
Chargé de Recherche HDR, École des Ponts ParisTech
Professeur, INP Bordeaux
Maître de Conférences, Université de Bordeaux
Maître de Conférences, INP Bordeaux

Président
Rapporteur
Rapporteur
Directeur de thèse
Co-encadrant
Co-encadrant

Titre : Estimation de paramètres évoluant sur des groupes de Lie : Application à la cartographie et à la localisation d'une caméra monoculaire

Résumé : Dans ce travail de thèse, nous proposons plusieurs algorithmes permettant d'estimer des paramètres évoluant sur des groupes de Lie. Ces algorithmes s'inscrivent de manière générale dans un cadre bayésien, ce qui nous permet d'établir une notion d'incertitude sur les paramètres estimés. Pour ce faire, nous utilisons une généralisation de la distribution normale multivariée aux groupes de Lie, appelée distribution normale concentrée sur groupe de Lie.

Dans une première partie, nous nous intéressons au problème du filtrage de Kalman à temps discret et continu-discret où l'état et les d'observations appartiennent à des groupes de Lie. Cette étude nous conduit à la proposition de deux filtres ; le CD-LG-EKF qui permet de résoudre un problème à temps continu-discret et le D-LG-EKF qui permet de résoudre un problème à temps discret.

Dans une deuxième partie, nous nous inspirons du lien entre optimisation et filtrage de Kalman, qui a conduit au développement du filtrage de Kalman étendu itéré sur espace euclidien, en le transposant aux groupes de Lie. Nous montrons ainsi comment obtenir une généralisation du filtre de Kalman étendu itéré aux groupes de Lie, appelée LG-IEKF, ainsi qu'une généralisation du lisseur de Rauch-Tung-Striebel aux groupes de Lie, appelée LG-RTS.

Finalement, dans une dernière partie, les concepts et algorithmes d'estimation sur groupes de Lie proposés dans la thèse sont utilisés dans le but de fournir des solutions au problème de la cartographie d'un environnement à partir d'une caméra monoculaire d'une part, et au problème de la localisation d'une caméra monoculaire se déplaçant dans un environnement préalablement cartographié d'autre part.

Mots clés : filtrage, lissage, variété, groupe de Lie, bayésien, pose de caméra, incertitude

Title : Parameter estimation on Lie groups: Application to mapping and localization from a monocular camera

Abstract : In this thesis, we derive novel parameter estimation algorithms dedicated to parameters evolving on Lie groups. These algorithms are casted in a Bayesian formalism, which allows us to establish a notion of uncertainty for the estimated parameters. To do so, a generalization of the multivariate normal distribution to Lie groups, called concentrated normal distribution on Lie groups, is employed.

In a first part, we generalize the Continuous-Discrete Extended Kalman Filter (CD-

EKF), as well as the Discrete Extended Kalman Filter (D-EKF), to the case where the state and the observations evolve on Lie groups. We obtain two novel algorithms called Continuous-Discrete Extended Kalman Filter on Lie Groups (CD-LG-EKF) and Discrete Extended Kalman Filter on Lie Groups (D-LG-EKF).

In a second part, we focus on bridging the gap between the formulation of intrinsic non linear least squares criteria and Kalman filtering/smoothing on Lie groups. We propose a generalization of the Euclidean Iterated Extended Kalman Filter (IEKF) to Lie groups, called LG-IEKF. We also derive a generalization of the Rauch-Tung-Striebel smoother (RTS), also known as Extended Kalman Smoother, to Lie groups, called LG-RTS.

Finally, the concepts and algorithms presented in the thesis are employed in a series of applications. Firstly, we propose a novel simultaneous localization and mapping approach. Secondly we develop an indoor camera localization framework. For this latter purpose, we derived a novel Rao-Blackwellized particle smoother on Lie groups, which builds upon the LG-IEKF and the LG-RTS.

Keywords : filtering, smoothing, manifold, Lie group, Bayesian, camera pose, uncertainty

Unité de recherche

Laboratoire IMS, CNRS UMR-5218, Groupe Signal et Image

Remerciements

Je tiens tout d'abord à remercier mes encadrants de thèse Audrey, Rémi et Yannick de m'avoir non seulement accordé leur confiance pendant ces trois années mais également pour m'avoir permis d'explorer les sujets qui m'intéressaient.

Un grand merci à Salem pour sa patience et sa disponibilité lorsqu'il s'agissait de répondre à toutes les questions que je lui posais. Et bien sûr, merci à Marc de m'avoir guidé sur le chemin périlleux des équations différentielles stochastiques sur groupe de Lie.

J'en viens maintenant à remercier Tic et Tac pour leur patience et leur amabilité surtout lorsque je devais remplir des tâches administratives dans lesquelles j'excelle particulièrement.

Voici maintenant une liste (non exhaustive) des pseudonymes des personnes m'ayant apporté la distraction nécessaire à tout travail de recherche : « la grande folle belge », « Émile, le chef du Cartel », « Doghrajipoulos », « Grosse konar », « le maître du Cactus », « le trafiquant de dossiers », « le receleur de bananes », « le charmeur de Béziers », « le membre de l'association des amis de Sarkozy », « le libriste » et « Pikachu » . Merci à vous tous !

Je remercie tout particulièrement Cornelia pour son aide durant ces trois années. Je dois avouer que cette thèse aurait été autrement plus compliquée si elle n'avait pas été là pour m'écouter et m'apporter ses conseils dans les (nombreux) moments de doute.

Finalement, je remercie ma famille pour le soutien qu'ils m'ont apporté tout au long de cette thèse et plus généralement tout au long de mes études.

*“Research is to see what everybody else has seen,
and to think what nobody else has thought.”*

Albert Szent-Györgyi

Table des matières

Liste des publications	1
Acronymes	3
Introduction	5
1 Introduction aux groupes de Lie matriciels	11
1.1 Groupes de Lie et Algèbres de Lie matriciels	12
1.1.1 Définitions	12
1.1.2 Applications exponentielle et logarithme d’une matrice .	14
1.1.3 Non commutativité	17
1.1.4 Formules de Baker-Campbell-Hausdorff	18
1.1.5 Dérivées première et seconde de l’application logarithme de matrice	20
1.1.6 Exemples de groupes de Lie	22
1.1.7 Transformation géométrique et interpolation	26
1.1.8 Détails d’implémentation	29
1.2 La distribution normale concentrée sur groupe de Lie matriciel .	31
1.2.1 Définition	31
1.2.2 Reparamétrisation de la covariance	35
1.2.3 Marginalisation	36
1.2.4 Cas particulier de la distribution normale multivariée . .	38
1.2.5 Cas particulier de la distribution log-normale	38
1.2.6 La distribution “en banane” sur $SE(2)$	39
1.3 Résumé des contributions et conclusion	40
2 Filtrage de Kalman à temps discret et continu-discret sur groupes de Lie matriciels	43
2.1 Introduction	44
2.2 État de l’art	45
2.3 Filtre de Kalman étendu continu-discret	47

2.3.1	Rappels sur le filtrage de Kalman étendu continu-discret sur espace euclidien	47
2.3.2	Modèles de propagation et d'observation proposés	49
2.3.3	Solution proposée	50
2.3.4	Propagation	50
2.3.5	Mise à jour	55
2.3.6	Algorithme CD-LG-EKF	58
2.3.7	Application du CD-LG-EKF à un problème de filtrage de la trajectoire d'une caméra	59
2.4	Filtre de Kalman étendu discret	71
2.4.1	Rappels sur le filtrage de Kalman étendu discret sur espace euclidien	71
2.4.2	Modèles de propagation et d'observation proposés	73
2.4.3	Solution proposée	73
2.4.4	Propagation	73
2.4.5	Mise à jour	75
2.4.6	Algorithme D-LG-EKF	76
2.4.7	Application du D-LG-EKF à un problème de filtrage de la trajectoire d'une caméra	76
2.5	Résumé des contributions et conclusion	80
3	De l'optimisation intrinsèque au filtrage et lissage de Kalman sur groupes de Lie matriciels	83
3.1	Introduction	85
3.2	État de l'art	86
3.3	Approximation de Gauss-Laplace intrinsèque	87
3.3.1	Problème	87
3.3.2	Solution proposée	87
3.4	Algorithme de Gauss-Newton intrinsèque	89
3.4.1	Introduction à l'optimisation intrinsèque	89
3.4.2	Algorithme LG-GN	90
3.5	Filtre de Kalman étendu itéré	91
3.5.1	Rappels sur le filtrage de Kalman étendu itéré sur espace euclidien	91
3.5.2	Modèles de propagation et d'observation proposés	94
3.5.3	Solution proposée	96
3.5.4	Propagation	96
3.5.5	Mise à jour	98
3.5.6	Algorithme LG-IEKF	102
3.5.7	Application du LG-IEKF au débruitage de transformations relatives	105

3.6	Lisseur de Rauch–Tung–Striebel	120
3.6.1	Rappels sur le lisseur de Rauch–Tung–Striebel sur espace euclidien	120
3.6.2	Modèle proposé	122
3.6.3	Solution proposée	123
3.6.4	Application du LG-GN	124
3.6.5	Algorithme LG-RTS	127
3.6.6	Application du LG-RTS sur un problème de lissage de la trajectoire d’une caméra	129
3.7	Résumé des contributions et conclusion	131
4	Cartographie et localisation à partir d’une caméra monoculaire	133
4.1	Introduction	133
4.2	Cartographie à partir d’une caméra monoculaire	134
4.2.1	Introduction	134
4.2.2	État de l’art	136
4.2.3	Architecture de VSLAM monoculaire proposée	138
4.2.4	Sélection d’images clés	138
4.2.5	Reconstruction d’une sous-carte	139
4.2.6	Estimation de la similitude 3D entre deux sous-cartes	142
4.2.7	Débruitage de similitudes 3D relatives	142
4.2.8	Limites de la méthode proposée	151
4.2.9	Résultats	152
4.3	Localisation à partir d’une caméra monoculaire dans un environnement cartographié	159
4.3.1	Introduction	159
4.3.2	État de l’art	163
4.3.3	Architecture du système de localisation en environnement intérieur	166
4.3.4	Un problème de suivi de cible sur $SE(3)$	168
4.3.5	Limites de la méthode proposée	176
4.3.6	Résultats	178
4.4	Résumé des contributions et conclusion	181
	Conclusion et perspectives	183
	Annexes	187
A	Marginalisation de la distribution normale concentrée sur groupe de Lie	191
B	Dérivées première et seconde de l’application logarithme de matrice	195

C	Propriétés de l'opérateur $\text{ad}_G(\cdot)$	199
D	Moyenne et covariance d'un système dynamique	201
E	Complétion du carré	203
F	Lemmes d'inversion matricielle	205
G	Démonstration du lisseur de Rauch–Tung–Striebel sur groupe de Lie	207
H	Démonstration du lisseur particulière Rao-Blackwellisé sur groupe de Lie	213
Bibliographie		223

Liste des publications

Revues internationales

“From Intrinsic Optimization to Iterated Extended Kalman Filtering on Lie Groups”

G.BOURMAUD, R.MÉGRET, A.GIREMUS, Y.BERTHOUMIEU

Journal of Mathematical Imaging and Vision (JMIV), 2015

“Continuous-Discrete Extended Kalman Filter on Matrix Lie Groups Using Concentrated Gaussian Distributions”

G.BOURMAUD, M.ARNAUDON, R.MÉGRET, A.GIREMUS

Journal of Mathematical Imaging and Vision (JMIV), 2015

Conférences et workshop internationaux

“Robust Wearable Camera Localization as a Target Tracking Problem on $SE(3)$ ”

G.BOURMAUD, A.GIREMUS

British Machine Vision Conference (BMVC), 2015

“Robust Large Scale Monocular Visual SLAM”

G.BOURMAUD, Y.BERTHOUMIEU

Computer Vision and Pattern Recognition (CVPR), 2015

“Global Motion Estimation from Relative Measurements in the Presence of Outliers”

G.BOURMAUD, R.MÉGRET, A.GIREMUS, Y.BERTHOUMIEU

Asian Conference on Computer Vision (ACCV), 2014

“Global Motion Estimation from Relative Measurements using Iterated Extended Kalman Filter on Matrix Lie Groups”

G.BOURMAUD, R.MÉGRET, A.GIREMUS, Y.BERTHOUMIEU

International Conference on Image Processing (ICIP), 2014

“Indoor Trajectory Estimation from Wearable Camera for Activity Monitoring”

G.BOURMAUD, R.MÉGRET, A.GIREMUS, Y.BERTHOUMIEU

3rd Workshop on Egocentric Vision, Computer Vision and Pattern Recognition (CVPR), 2014

“Discrete Extended Kalman Filter on Lie Groups”

G.BOURMAUD, R.MÉGRET, A.GIREMUS, Y.BERTHOUMIEU

European Signal Processing Conference (EUSIPCO), 2013

Conférences nationales

“Un Lisseur de Rauch-Tung-Striebel sur Groupe de Lie”

G.BOURMAUD

Groupe d'Etudes du Traitement du Signal et des Images (GRETSI), 2015

“Filtre de Kalman Etendu Discret sur Groupe de Lie”

G.BOURMAUD, R.MÉGRET, A.GIREMUS, Y.BERTHOUMIEU

Groupe d'Etudes du Traitement du Signal et des Images (GRETSI), 2013

Acronymes

Nous listons ici les différents acronymes utilisés dans ce mémoire :

BA	Algorithme d'ajustement de faisceaux (Bundle Adjustment)
BCH	Baker-Campbell-Hausdorff
CBIR	Algorithme de recherche par le contenu (Content Based Image Retrieval)
CD-Constr-NLF	Filtre non linéaire continu-discret avec une contrainte extrinsèque (Continuous-Discrete Constrained Non Linear Filter)
CD-Constr-UKF	Filtre de Kalman sans parfum continu-discret avec une contrainte extrinsèque (Continuous-Discrete Constrained Unscented Kalman Filter)
CD-EKF	Filtre de Kalman étendu continu-discret (Continuous-Discrete Extended Kalman Filter)
CD-LG-EKF	Filtre de Kalman étendu continu-discret sur groupe de Lie (Continuous-Discrete Extended Kalman Filter on Lie Groups)
D-LG-EKF	Filtre de Kalman étendu discret sur groupe de Lie (Discrete Extended Kalman Filter on Lie Groups)
EKF	Filtre de Kalman étendu (Extended Kalman Filter)
EM	Algorithme Espérance-Maximisation
GN	Algorithme de Gauss-Newton
IEKF	Filtre de Kalman itéré étendu (Iterated Extended Kalman Filter)
LBP	Algorithme de propagation de croyance (Loopy Belief Propagation)
LG-EKF	Filtre de Kalman étendu sur groupe de Lie (Extended Kalman Filter on Lie Groups)

LG-GN	Algorithme de Gauss-Newton intrinsèque sur groupe de Lie
LG-IEKF	Filtre de Kalman itéré étendu sur groupe de Lie (Iterated Extended Kalman Filter on Lie Groups)
LP	Algorithme d'optimisation linéaire (Linear Program)
LG-RBPS	Lisseur Rao-Blackwellisé sur groupe de Lie (Rao-Blackwellized Particle Smoother on Lie Groups)
LG-RTS	Lisseur de Rauch-Tung-Striebel sur groupe de Lie
LS-RSA	Algorithme de débruitage de similitudes 3D relatives (Large Scale Relative Similarity Averaging)
Pol	Point d'intérêt (Point Of Interest)
RBPF	Filtre Rao-Blackwellisé (Rao-Blackwellized Particle Filter)
RBPS	Lisseur Rao-Blackwellisé (Rao-Blackwellized Particle Smoother)
RMSE	Racine carrée de la moyenne des erreurs quadratiques (Root Mean Square Error)
RTS	Lisseur de Rauch-Tung-Striebel
SLAM	Problème de cartographie et de localisation simultanée (Simultaneous Localization And Mapping)
UKF	Filtre de Kalman sans parfum (Unscented Kalman Filter)
VIL	Problème de localisation par caméra en environnement intérieur (Visual Indoor Localization)
VO	Odométrie Visuelle (Visual Odometry)
VSLAM	SLAM à partir d'une caméra (Visual SLAM)

Introduction

Contexte

Le travail présenté dans ce manuscrit s'inscrit dans le cadre du projet européen Dem@Care (septième programme cadre de la Communauté européenne, FP7/2007-2013, subvention 288199) dont l'objectif était de fournir aux médecins des informations pouvant les aider à diagnostiquer l'état d'avancement de la maladie d'Alzheimer chez un patient donné.

Afin de fournir des informations pertinentes aux médecins, des capteurs (caméra portée par le patient, caméra fixe, accéléromètres, microphones, etc.) sont placés dans l'environnement où le patient s'adonne à des activités de la vie quotidienne, et les signaux issus de ces capteurs sont traités automatiquement par des programmes informatiques.

Parmi les informations qu'il est important d'extraire, la localisation du patient est une donnée pouvant être exploitée, soit directement par le médecin, soit en entrée d'algorithmes de fusion de données visant à obtenir des informations plus facilement interprétables telles que la nature de l'activité concernée ou encore sa durée. C'est dans ce contexte que nous avons été amenés à travailler sur le problème de la localisation d'une personne en environnement intérieur.

Le flux vidéo issu de la caméra portée par le patient, qui présente un point de vue privilégié sur l'action effectuée par ce dernier, constitue une source d'informations importante permettant de déterminer la localisation du patient dans l'environnement. Ainsi, nous nous sommes intéressés au problème de la localisation d'une caméra portée évoluant dans un environnement intérieur.

Le problème que nous souhaitons résoudre pouvait donc être vu comme un problème d'estimation de la position et de l'orientation d'une caméra, également appelé pose de la caméra, à partir de son flux vidéo.

Ce flux vidéo étant acquis séquentiellement, notre travail s'est naturellement orienté vers des approches de filtrage et de lissage permettant d'estimer récursivement la pose de la caméra.

Cependant, une caractéristique importante de la pose d'une caméra réside

dans le fait que cette dernière évolue, non pas sur un espace euclidien, mais sur une variété différentiable appelée groupe de Lie. C'est ainsi que nous avons choisi d'étudier le problème de l'estimation de paramètres évoluant sur un groupe de Lie.

Plutôt que de nous intéresser à des approches de type filtrage ou lissage particulière, dont le coût calculatoire peut sensiblement augmenter lorsque les paramètres à estimer évoluent sur une variété, nous avons préféré développer des approches analytiques. Plus particulièrement, nous nous sommes intéressés à l'extension du filtre et du lisseur de Kalman, qui permettent d'estimer un état évoluant sur un espace euclidien à partir d'observations elles-mêmes euclidiennes, aux groupes de Lie.

Motivation

L'étude des variétés différentiables appliquée au traitement du signal et des images, à la robotique ou encore à la vision par ordinateur a connu un véritable essor ces dernières années. Il est en effet courant, dans ces différents domaines, que la formulation de problèmes d'estimation de paramètres conduise à des contraintes géométriques sur ces derniers. Dans bon nombre de cas, ces contraintes impliquent que les paramètres inconnus appartiennent alors à une variété différentiable, généralement Riemannienne, parmi lesquelles on trouve les variétés de Stiefel, de Grassmann, ou encore les groupes de Lie.

Dans cette thèse, nous avons choisi de nous intéresser au problème de l'estimation de paramètres évoluant sur un groupe de Lie, et plus spécifiquement évoluant sur un groupe de Lie matriciel¹. Deux raisons expliquent ce choix : en premier lieu l'élégance de la structure géométrique sous-jacente aux groupes de Lie et en second lieu leur implication dans de nombreux problèmes présents en vision par ordinateur.

Un élément d'un groupe de Lie (matriciel) peut être vu comme une matrice carrée soumise à diverses contraintes qui déterminent la géométrie de la variété différentiable. Prenons par exemple le cas des matrices de rotation en dimension 3 qui forment le groupe de Lie Spécial Orthogonal de dimension 3 ($SO(3)$). Un élément X de ce groupe de Lie est une matrice carrée de taille 3 telle que $X^T X = Id$ et $\det(X) = 1$. Ainsi, les matrices appartenant à $SO(3)$ sont formées de 9 éléments mais la dimension du groupe de Lie (on parle également du nombre de degrés de liberté) est 3 en raison des six contraintes qui lui sont imposées. Une manière intuitive de se représenter ces 3 degrés de liberté est d'essayer de faire pivoter une maquette d'avion. Ainsi, on se rend compte que les 3 axes de rotations, qui nous apparaissent naturellement (appelés lacet, tangage

1. La plupart des groupes de Lie rencontrés en physique sont des groupes de Lie matriciels [Gilmore 2008].

et roulis), suffisent à faire pivoter la maquette dans toutes les orientations possibles.

Cette distinction entre le nombre d'éléments constituant une matrice évoluant sur un groupe de Lie et le nombre de degrés de liberté de ce groupe de Lie nous conduit à une série de questions fondamentales lorsqu'on s'intéresse à un groupe de Lie :

Quelle est la distance entre deux éléments d'un groupe de Lie ?

Quel est le chemin le plus court entre deux éléments d'un groupe de Lie ?

Comment se déplace-t-on sur un groupe de Lie ?

Les réponses à ces questions paraissent évidentes lorsqu'on manipule des éléments appartenant à un espace euclidien. En effet, ce dernier est un espace vectoriel ce qui nous permet de nous déplacer, il est muni d'un produit scalaire ce qui nous permet de définir la distance euclidienne, et le chemin le plus court entre deux points est la ligne droite passant par ces deux points. Cette géométrie euclidienne est au cœur des méthodes d'estimation de paramètres euclidiens telles que le filtre de Kalman.

En revanche, lorsqu'on s'intéresse aux groupes de Lie, il devient clair qu'il n'existe pas *une* bonne réponse à ces questions mais *différentes* réponses en vertu du fait qu'"Une géométrie ne peut pas être plus vraie qu'une autre ; elle peut seulement être plus commode." [Poincaré 1923]. C'est pourquoi nous sommes conduits à discuter des différentes géométries que l'on peut considérer et tenter de motiver notre choix en expliquant en quoi celui-ci est plus *commode* que les autres, notamment du point de vue de l'estimation de paramètres.

Une première géométrie consiste à *paramétrer* la variété. Cela revient à transformer les paramètres que l'on cherche à estimer, qui sont contraints à vivre sur un groupe de Lie, en de nouveaux paramètres évoluant sur un espace euclidien et dont le nombre est égal au nombre de degrés de liberté du groupe de Lie. Dans le cas des matrices de rotation en dimension 3, les angles de lacet, de tangage et de roulis sont appelés les angles d'Euler et définissent une paramétrisation possible du groupe de Lie. De cette manière, on ne cherche plus à estimer 9 éléments partageant des contraintes mais 3 paramètres euclidiens. Ainsi, cette géométrie a l'avantage de permettre l'utilisation de l'ensemble des méthodes d'estimation euclidiennes. Par exemple, le filtre et le lisseur de Kalman s'appliquent directement. Cependant, en règle générale, il n'existe pas d'isomorphisme entre un groupe de Lie et un espace euclidien. Par conséquent, le fait de *paramétrer* le groupe de Lie produit des *singularités* qui peuvent fortement dégrader l'estimation des paramètres. C'est par exemple le cas des angles d'Euler dont la singularité créée porte le nom de *blocage de cardan* et a conduit les scientifiques à se détourner des angles d'Euler pour l'estimation d'une rotation.

Une deuxième géométrie consiste à voir les paramètres que l'on souhaite estimer comme étant à valeur dans l'espace des matrices carrées, mais devant respecter certaines contraintes. Dans ce cas, la distance considérée entre deux éléments d'un groupe de Lie est essentiellement la distance euclidienne entre les deux matrices. Cette géométrie permet de replacer le problème d'estimation de paramètres évoluant sur un groupe de Lie dans un contexte plus général d'estimation sous contraintes. Par conséquent, il est possible de recourir à l'ensemble des techniques d'estimation sous contraintes. Une manière d'appliquer un filtre de Kalman dans le cadre de cette géométrie revient à appliquer le filtre dans l'espace des matrices carrées puis à projeter l'estimé sur la variété. Cependant, l'espace considéré est de plus grande dimension que celle du groupe de Lie considéré et les algorithmes sont plus facilement sujets aux mauvais conditionnements. En poursuivant notre exemple sur les matrices de rotation en dimension 3, cette approche reviendrait à estimer un vecteur euclidien de dimension 9 sous diverses contraintes retranscrivant la contrainte d'orthonormalité d'une matrice de rotation.

Une troisième géométrie, celle dans laquelle cette thèse s'inscrit, revient à prendre en compte, de manière *intrinsèque* le fait que les paramètres que l'on cherche à estimer vivent sur un groupe de Lie. Plus intuitivement, toujours en utilisant l'exemple des matrices de rotation en dimension 3, cela signifie que la matrice de rotation estimée avec une méthode utilisant cette géométrie est bien de taille 3×3 et respecte bien la contrainte d'orthonormalité. Cependant, la méthode d'estimation n'a pas "conscience qu'à côté" de cette matrice de rotation il y a d'autres matrices de taille 3×3 qui ne respectent pas la contrainte d'orthonormalité. Pour cette méthode, seules les matrices de rotation existent. On dit que l'espace de recherche est *intrinsèquement* contraint. L'intérêt d'une telle géométrie réside dans son élégance, et le fait que les méthodes d'estimation ont alors de *bonnes* propriétés de convergence et une grande stabilité numérique.

Au sein de cette troisième approche, il existe en réalité plusieurs géométries, c'est à dire que plusieurs manières de contraindre *intrinsèquement* l'espace de recherche sont envisageables. Une première manière revient à s'appuyer sur la structure riemannienne du groupe de Lie (car un groupe de Lie est une variété riemannienne). Une seconde serait de considérer une structure venant de l'algèbre géométrique. Une troisième repose sur la notion de *sous-groupe à un paramètre*. Du point de vue du groupe de Lie $SO(3)$, utiliser une structure de l'algèbre géométrique revient à employer le formalisme des quaternions unitaires qui forment un double recouvrement de $SO(3)$. En revanche, pour le cas spécifique de $SO(3)$, l'utilisation de la structure riemannienne ou bien celle du *sous-groupe à un paramètre* ne change rien car le chemin le plus court entre deux éléments est le même. Cette propriété n'est pas vérifiée pour tout groupe

de Lie et ne l'est pas pour les principaux groupes de Lie que nous considérons dans nos applications.

Justifier le choix d'une géométrie plutôt qu'une autre semble difficile à ce stade. Néanmoins, comme nous tentons de le montrer dans cette thèse, le fait d'exploiter les propriétés du *sous-groupe à un paramètre* est *commode* et naturel puisque cette géométrie permet des généralisations à la fois élégantes et calculatoirement efficaces d'algorithmes d'estimation euclidiens, tels que le filtre et le lisseur de Kalman. De plus, nous montrons sur divers problèmes de vision par ordinateur que ces algorithmes permettent d'obtenir de meilleurs résultats que ceux jusqu'ici répertoriés dans l'état de l'art.

Contenu du manuscrit

Nous consacrons le premier chapitre de cette thèse à la présentation des principaux outils théoriques et pratiques utilisés tout au long du manuscrit afin d'estimer des paramètres évoluant sur des groupes de Lie matriciels. Les concepts introduits sont réunis ici afin de les présenter de manière intuitive en proposant des illustrations graphiques permettant d'interpréter géométriquement ces notions souvent abstraites.

Dans un second chapitre, nous nous intéressons au développement de filtres de Kalman dédiés à l'estimation de paramètres vivant sur un groupe de Lie matriciel à partir d'observations appartenant elles-mêmes à un (autre) groupe de Lie matriciel. Nous proposons deux filtres, qui constituent la principale contribution de ce chapitre, et permettent de résoudre respectivement un problème à temps continu-discret et un problème à temps discret. De plus, ces deux filtres ont la propriété de se réduire aux filtres de Kalman étendu discret et continu-discret lorsque l'état et les observations évoluent sur des espaces euclidiens.

Le troisième chapitre est consacré au lien que l'on peut établir entre l'optimisation sur groupe de Lie d'un côté et le filtrage/lissage de Kalman sur groupe de Lie d'un autre côté. Cette analyse permet notamment d'interpréter les algorithmes proposés dans le second chapitre sous un nouvel angle. Nous obtenons ainsi une généralisation du filtre de Kalman itéré étendu aux groupes de Lie ainsi qu'une généralisation du lisseur de Rauch-Tung-Striebel (également connu sous le nom de lisseur de Kalman étendu) aux groupes de Lie.

Les concepts et algorithmes décrits dans les chapitres précédents sont mis à profit dans un quatrième chapitre dans le but de proposer des solutions au problème de la cartographie d'un environnement à partir d'une caméra monoculaire d'une part, et au problème de la localisation d'une caméra monoculaire se déplaçant dans un environnement préalablement cartographié d'autre part. Nous proposons notamment un lisseur Rao-Blackwellisé sur groupe de Lie

reposant sur un banc de filtres et lisseurs de Kalman étendus sur groupe de Lie. Cet algorithme nous permet d'estimer la localisation d'une caméra à partir de son flux vidéo.

Nous terminons le manuscrit par une conclusion et des perspectives de recherches futures.

Introduction aux groupes de Lie matriciels

1.1	Groupes de Lie et Algèbres de Lie matriciels . . .	12
1.1.1	Définitions	12
1.1.2	Applications exponentielle et logarithme d'une matrice	14
1.1.3	Non commutativité	17
1.1.4	Formules de Baker-Campbell-Hausdorff	18
1.1.5	Dérivées première et seconde de l'application logarithme de matrice	20
1.1.6	Exemples de groupes de Lie	22
1.1.7	Transformation géométrique et interpolation	26
1.1.8	Détails d'implémentation	29
1.2	La distribution normale concentrée sur groupe de Lie matriciel	31
1.2.1	Définition	31
1.2.2	Reparamétrisation de la covariance	35
1.2.3	Marginalisation	36
1.2.4	Cas particulier de la distribution normale multivariée	38
1.2.5	Cas particulier de la distribution log-normale	38
1.2.6	La distribution "en banane" sur $SE(2)$	39
1.3	Résumé des contributions et conclusion	40

Ce chapitre est consacré à la présentation des principaux outils théoriques et pratiques que nous utilisons tout au long du manuscrit afin d'estimer des paramètres évoluant sur des groupes de Lie matriciels.

1.1 Groupes de Lie et Algèbres de Lie matriciels

Nous proposons ici une introduction au formalisme des groupes de Lie. Ce sous-chapitre ne saurait remplacer un ouvrage dédié à ces derniers, c’est pourquoi nous renvoyons le lecteur à [Abbaspour 2007 ; Hall 2003 ; Chirikjian 2012] pour une description plus détaillée.

Dans ce manuscrit, nous nous intéressons aux groupes de Lie matriciels car la plupart des groupes de Lie rencontrés en physique sont des groupes de matrices [Gilmore 2008]. De plus, la majorité des groupes de Lie de dimension finie sont isomorphes à des groupes de Lie matriciels en raison du fait que toute algèbre de Lie est isomorphe à une algèbre matricielle (d’après le théorème d’Ado [Ado 1947]).

1.1.1 Définitions

Un groupe de Lie G est un groupe muni d’une structure de variété différentiable de telle sorte que les opérations de multiplication et d’inversion du groupe sont différentiables. Pour une description détaillée des variétés différentiables, le lecteur peut consulter [Kolar 1993]. Si G est un groupe de Lie matriciel, alors $G \subset \mathbb{R}^{n \times n}$ et ses opérations de multiplication et inversion sont simplement la multiplication matricielle et l’inversion matricielle. Par conséquent, l’élément identité du groupe correspond à la matrice identité Id .

Dans ce contexte, il est possible de définir l’application “action à gauche”¹ $L_Y : G \rightarrow G$ du groupe de Lie sur lui-même :

$$L_Y X = YX \tag{1.1}$$

où $Y, X \in G$. Puisque G est une variété différentiable, il est possible d’attacher à tout point $X \in G \subset \mathbb{R}^{n \times n}$ un espace tangent $T_X G \subset \mathbb{R}^{n \times n}$. Cet espace tangent est un espace vectoriel dont la dimension p est appelé dimension de la variété.

Remarquons que la dimension p du groupe de Lie est définie comme sa dimension au sens de sa structure de variété différentielle, alors que n représente la dimension des matrices carrées qui permettent une représentation matricielle de G . Le groupe de Lie $SL(3)$, par exemple, qui est présenté plus en détails dans la suite de ce chapitre, est un groupe de Lie de dimension $p = 8$ dont les éléments peuvent être représentés par des matrices carrées de taille $n = 3$.

1. Notons qu’il est possible de définir de manière équivalente l’application “action à droite”.

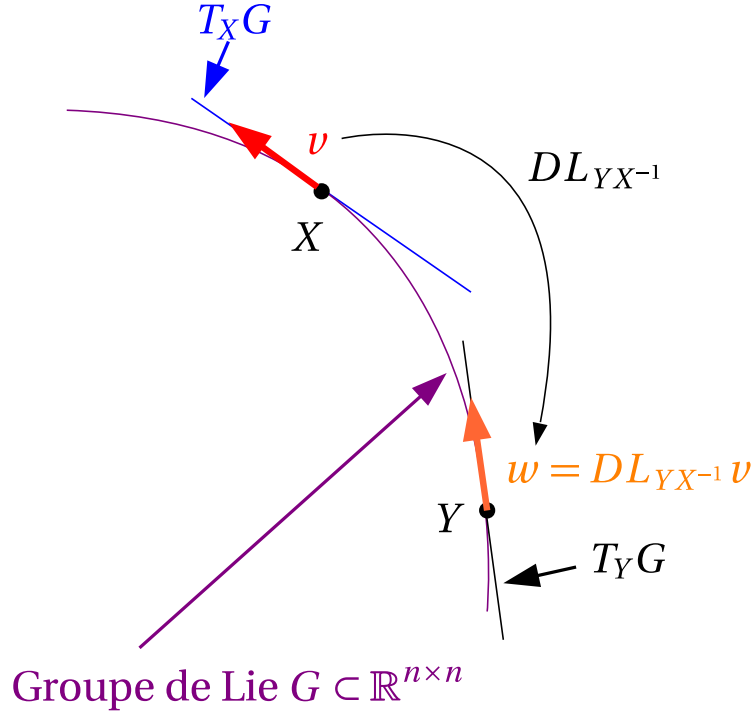


Figure 1.1 – Illustration graphique de l'application tangente

En utilisant l'action à gauche du groupe sur lui même L_Y , on peut définir une application tangente (à gauche)

$$DL_Y : T_X G \rightarrow T_Y G \quad (1.2)$$

avec

$$DL_Y v = Yv \quad (1.3)$$

où $v \in T_X G \subset \mathbb{R}^{n \times n}$. L'action à gauche permet donc de transporter un vecteur v appartenant à l'espace tangent $T_X G$ en n'importe quel autre espace tangent. Une illustration graphique de l'application tangente est proposée en figure 1.1.

Cette propriété nous permet d'introduire la notion de *champ de vecteurs invariant à gauche* qui associe un vecteur tangent $v(X) \in T_X G$ à chaque élément X du groupe en transportant $v(Id)$, le vecteur défini dans l'espace tangent à l'identité du groupe, sur tout le groupe en utilisant l'application tangente DL_X :

$$v(X) = DL_X v(Id) \quad (1.4)$$

Ainsi, un *champs de vecteurs invariant à gauche* dépend uniquement de $v(Id)$. C'est pourquoi, dans le formalisme des groupes de Lie, une attention particulière est accordée à l'espace tangent à l'identité du groupe $T_{Id} G$ que l'on appelle *l'algèbre de Lie* et que l'on note \mathfrak{g} .

L'algèbre de Lie $\mathfrak{g} \subset \mathbb{R}^{n \times n}$ associée à un groupe de Lie matriciel $G \subset \mathbb{R}^{n \times n}$

de dimension p est par conséquent un espace vectoriel de dimension p . Il est alors possible de définir une base de matrices réelles de l'algèbre de Lie $E_i \in \mathbb{R}^{n \times n}$ pour $i = 1 \dots p$.

D'un point de vue numérique, il est plus simple et efficace de travailler avec des vecteurs colonnes de taille p plutôt qu'avec des matrices de taille $n \times n$. Par conséquent, nous définissons un isomorphisme entre \mathfrak{g} et \mathbb{R}^p :

$$\begin{aligned} [\cdot]_G^\vee : \mathfrak{g} &\rightarrow \mathbb{R}^p \\ [\cdot]_G^\wedge : \mathbb{R}^p &\rightarrow \mathfrak{g} \end{aligned} \quad (1.5)$$

Exemple : Prenons $\mathbf{a} \in \mathfrak{g} \subset \mathbb{R}^{n \times n}$, alors $[\mathbf{a}]_G^\vee = \mathbf{a} \in \mathbb{R}^p$. Ainsi $[E_i]_G^\vee = e_i$ où $\{e_i\}_{i=1 \dots p}$ est la base naturelle de \mathbb{R}^p et $\mathbf{a} = \sum_{i=1}^p a_i E_i$ avec $\mathbf{a} = (a_1, \dots, a_p)^T$.

1.1.2 Applications exponentielle et logarithme d'une matrice

Une manière d'exprimer le lien entre un groupe de Lie (matriciel) et son algèbre de Lie est de considérer la notion de *sous groupe à un paramètre* qui est défini ci-après. Considérons l'équation différentielle suivante qui représente l'évolution de $X(t) \in G$ en partant de $X(0) = Id$ avec une vitesse constante $\mathbf{a} \in \mathfrak{g}$:

$$\begin{cases} \dot{X}(t) = DL_{X(t)} \mathbf{a} = X(t) \mathbf{a} \\ X(0) = Id \end{cases} \quad (1.6)$$

Par définition $\dot{X}(t) \in T_{X(t)}G$. La solution de cette équation différentielle est donnée par :

$$X(t) = \exp_G(\mathbf{a}t) \quad (1.7)$$

où

$$\exp_G : \mathfrak{g} \rightarrow G \quad (1.8)$$

est l'application exponentielle de matrice de taille $n \times n$. L'application exponentielle de matrice est définie par la série entière :

$$\exp_G(\mathbf{a}) = \sum_{k=0}^{\infty} \frac{1}{k!} \mathbf{a}^k = Id + \mathbf{a} + \frac{1}{2} \mathbf{a}^2 + \frac{1}{6} \mathbf{a}^3 + \dots \quad (1.9)$$

Cette application permet donc de transformer un vecteur défini dans l'algèbre de Lie \mathfrak{g} en un élément du groupe de Lie G . Ainsi, l'application $\exp_G(\cdot)$ est bien un *sous groupe à un paramètre* de G puisqu'elle vérifie :

$$\exp_G(\mathbf{a}t_1) \exp_G(\mathbf{a}t_2) = \exp_G(\mathbf{a}(t_1 + t_2)) \quad (1.10)$$

Remarquons également que $\exp_G(\cdot)$ respecte la propriété suivante :

$$\exp_G(\mathbf{a}t)^{-1} = \exp_G(-\mathbf{a}t) \quad (1.11)$$

Il est intéressant de noter que les courbes engendrées par l'exponentielle de matrice sont des géodésiques au sens d'une connexion de Cartan-Schouten [Nielsen 2013]. Nous ne détaillons pas plus cet aspect puisque, dans le reste du manuscrit, nous n'exploiterons pas *explicitement* les propriétés de ces connexions.

La fonction (localement) inverse de l'application exponentielle de matrice est le logarithme de matrice, défini au voisinage de l'identité par la série entière (voir démonstration dans [Faraut 2008] Théorème 2.2.1) :

$$\log_G(X) = \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k} (X - Id)^k \quad (1.12)$$

Une illustration graphique des applications exponentielle et logarithme de matrice dans le contexte des groupes de Lie est proposée en figure 1.2. Notons que les applications exponentielle et logarithme de matrice n'établissent qu'un difféomorphisme local entre un voisinage ouvert de $\mathbf{0}_{n \times n}$ dans \mathfrak{g} et un voisinage ouvert de Id dans G . Par conséquent, nous définissons $\mathfrak{s} \subset \mathfrak{g}$ et $M \subset G$ comme les ensembles sur lesquels \exp_G et \log_G sont des fonctions réciproques. Plus précisément nous choisissons \mathfrak{s} comme étant un sous-ensemble ouvert, symétrique et connecté de \mathfrak{g} .

Dans la suite de ce manuscrit, lorsque nous employons les fonctions $\exp_G(\mathbf{a})$ et $\log_G(X)$, nous supposons que $\mathbf{a} \in \mathfrak{s}$, $X \in M$ et que les séries entières de \exp_G et \log_G convergent sur \mathfrak{s} et M respectivement. Ces notions sont résumées dans la figure 1.3 où on note par abus de langage $S = [\mathfrak{s}]_G^\vee \subset \mathbb{R}^p$. Le logarithme de matrice respecte notamment la propriété suivante :

$$\log_G(X^{-1}) = -\log_G(X) \quad (1.13)$$

Nous introduisons finalement les notations suivantes, qui permettent de travailler avec des vecteurs colonnes et donc de simplifier l'écriture des calculs à venir :

$$\exp_G^\wedge(a) = \exp_G([a]_G^\wedge) \quad (1.14)$$

et

$$\log_G^\vee(X) = [\log_G(X)]_G^\vee \quad (1.15)$$

où $a \in S$ et $X \in M$. Par abus de langage, le vecteur a dans l'équation (1.14) sera souvent appelé "élément de l'algèbre de Lie \mathfrak{g} " alors qu'il appartient en réalité à \mathbb{R}^p , l'espace euclidien relié à \mathfrak{g} par l'isomorphisme (1.5).

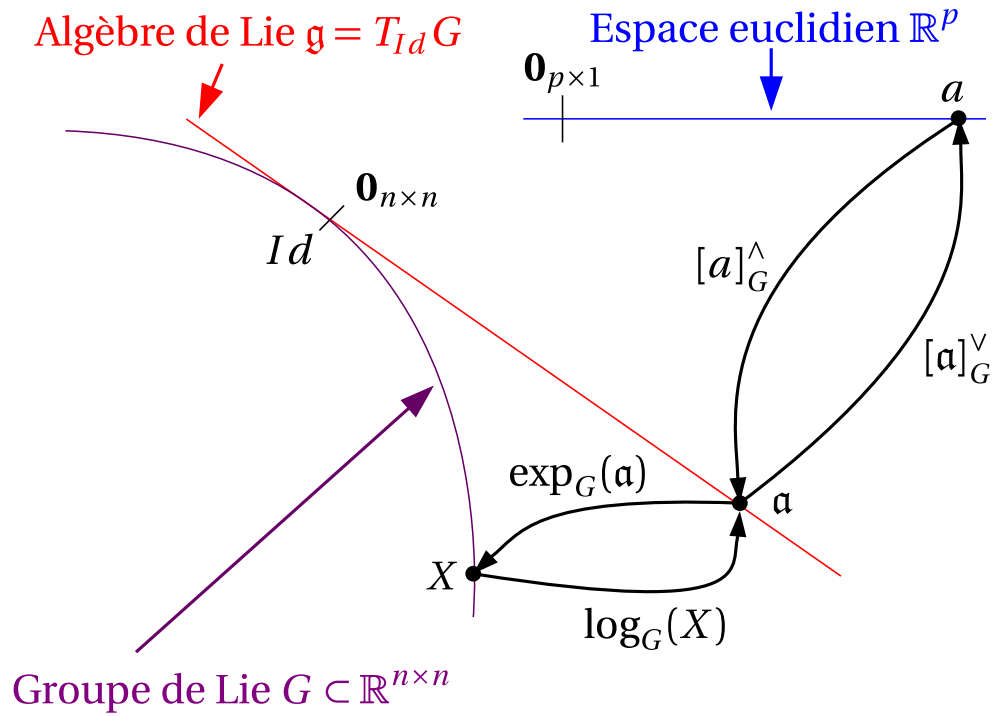


Figure 1.2 – Illustration graphique des applications exponentielle et logarithme d'une matrice dans le cadre d'un groupe de Lie

$ \begin{array}{ccc} M \subset G \subset \mathbb{R}^{n \times n} & \xrightarrow{\log_G} & \mathfrak{s} \subset \mathfrak{g} \subset \mathbb{R}^{n \times n} & \xrightarrow{[\cdot]_G^\vee} & S \subset [\mathfrak{g}]_G^\vee \subset \mathbb{R}^p \\ & \xleftarrow{\exp_G} & & \xleftarrow{[\cdot]_G^\wedge} & \end{array} $
--

Figure 1.3 – Résumé des notions reliant le groupe de Lie à son algèbre de Lie

1.1.3 Non commutativité

Les groupes de Lie matriciels sont en général non commutatifs :

$$X \exp_G(\mathfrak{a}) \neq \exp_G(\mathfrak{a}) X \quad (1.16)$$

En effet, l'action à droite de l'incrément $\exp_G(\mathfrak{a})$ sur X n'a pas le même effet que l'action à gauche. C'est là qu'intervient la représentation adjointe de G sur \mathfrak{g} , notée $\text{Ad}_G(\cdot)$, que nous définissons maintenant. Nous souhaitons trouver \mathfrak{b} de manière à satisfaire :

$$X \exp_G(\mathfrak{a}) = \exp_G(\mathfrak{b}) X \quad (1.17)$$

Nous obtenons alors :

$$\begin{aligned} \exp_G(\mathfrak{b}) &= X \exp_G(\mathfrak{a}) X^{-1} = X \sum_{k=0}^{\infty} \frac{1}{k!} \mathfrak{a}^k X^{-1} \\ &= \sum_{k=0}^{\infty} \frac{1}{k!} (X \mathfrak{a} X^{-1})^k = \exp_G(X \mathfrak{a} X^{-1}) \end{aligned} \quad (1.18)$$

car

$$(X \mathfrak{a} X^{-1})^k = X \mathfrak{a} X^{-1} X \mathfrak{a} X^{-1} \dots X \mathfrak{a} X^{-1} = X \mathfrak{a}^k X^{-1} \quad (1.19)$$

Finalement $\mathfrak{b} = X \mathfrak{a} X^{-1}$. En exprimant ce résultat dans \mathbb{R}^p ($b = [\mathfrak{b}]_G^\vee$ et $a = [\mathfrak{a}]_G^\vee$), nous avons :

$$b = [X \mathfrak{a} X^{-1}]_G^\vee = \text{Ad}_G(X) a \quad (1.20)$$

où $\text{Ad}_G(\cdot) \subset \mathbb{R}^{p \times p}$ est la représentation adjointe de G sur \mathfrak{g} qui traduit la non commutativité du groupe :

$$X \exp_G^\wedge(a) = \exp_G^\wedge(\text{Ad}_G(X) a) X \quad (1.21)$$

Il existe une autre représentation adjointe qui traduit également la non-commutativité d'un groupe de Lie. En effet, en prenant la dérivée directionnelle de $X \mathfrak{a} X^{-1}$, où nous définissons au préalable $X = \exp_G(\mathfrak{c}) = \exp_G^\wedge(c)$, nous faisons apparaître le *crochet de Lie* noté $[\cdot, \cdot]$ qui n'est autre que le commutateur de matrices :

$$\left. \frac{d \exp_G(s\mathfrak{c}) \mathfrak{b} \exp_G(-s\mathfrak{c})}{ds} \right|_{s=0} = \mathfrak{c}\mathfrak{b} - \mathfrak{b}\mathfrak{c} = [\mathfrak{c}, \mathfrak{b}] \quad (1.22)$$

Remarquons que le *crochet de Lie* $(\mathfrak{c}, \mathfrak{b}) \mapsto [\mathfrak{c}, \mathfrak{b}]$ de $\mathfrak{g} \times \mathfrak{g}$ dans \mathfrak{g} est anticommutatif et bilinéaire et vérifie les propriétés suivantes :

$$\forall \mathfrak{a} \in \mathfrak{g}, [\mathfrak{a}, \mathfrak{a}] = 0 \quad (1.23)$$

et

$$\forall \mathfrak{a}, \mathfrak{b}, \mathfrak{c} \in \mathfrak{g}, [\mathfrak{a}, [\mathfrak{b}, \mathfrak{c}]] + [\mathfrak{b}, [\mathfrak{c}, \mathfrak{a}]] + [\mathfrak{c}, [\mathfrak{a}, \mathfrak{b}]] = 0 \quad (1.24)$$

Puisque $[\cdot, \cdot]$, $[\cdot]_G^\vee$ et $[\cdot]_G^\wedge$ sont des fonctions linéaires, le *crochet de Lie* peut être exprimé dans \mathbb{R}^p de la manière suivante :

$$[[\mathfrak{c}, \mathfrak{b}]]_G^\vee = \left[[[c]_G^\wedge, [b]_G^\wedge] \right]_G^\vee = \text{ad}_G(c) b \quad (1.25)$$

où $\text{ad}_G : \mathbb{R}^p \rightarrow \mathbb{R}^{p \times p}$ est appelé représentation adjointe de \mathfrak{g} . Remarquons que dans le cas d'un groupe de Lie commutatif alors $\text{ad}_G(c) = \mathbf{0}_{p \times p}$, $\forall c \in \mathbb{R}^p$. Nous introduisons également la notation suivante où $(\text{ad}_G(c))_{ij}$ correspond à l'élément présent à la $i^{\text{ème}}$ ligne et $j^{\text{ème}}$ colonne de la matrice $\text{ad}_G(c)$:

$$(\text{ad}_G(c))_{ij} = L_{ij}^T c \quad (1.26)$$

où $L_{ij} \in \mathbb{R}^p$.

Nous terminons ce paragraphe sur la non-commutativité des groupes de Lie en énonçant la propriété suivante :

$$\text{Ad}_G(\exp_G^\wedge(a)) = \text{expm}(\text{ad}_G(a)) \quad (1.27)$$

pour tout $a \in \mathbb{R}^p$ avec $\text{expm}(\cdot)$ l'exponentielle de matrice d'une matrice de taille $p \times p$. Cette propriété peut être prouvée en utilisant l'équation (1.22).

1.1.4 Formules de Baker-Campbell-Hausdorff

Les formules de Baker-Campbell-Hausdorff (BCH) permettent d'exprimer le produit matriciel de deux éléments d'un groupe de Lie directement dans son algèbre de Lie. Ces expressions seront employées lorsque nous aurons besoin d'effectuer des *linéarisations*. Remarquons que la notion de “formule BCH” est étroitement liée à la non-commutativité des groupes de Lie. En effet, si un groupe de Lie est commutatif alors :

$$\log_G^\vee(\exp_G^\wedge(a) \exp_G^\wedge(b)) = a + b \quad (1.28)$$

Cependant, ça n'est généralement pas le cas.

1.1.4.1 Première formule BCH

D'après l'expression de la dérivée de l'exponentielle de matrice ([Faraut 2008], Théorème 2.1.4), nous obtenons le développement de Taylor à l'ordre 1 suivant :

$$\log_G^\vee (\exp_G^\wedge (-a) \exp_G^\wedge (a+b)) = \Phi_G (-a) b + O(\|b\|^2) \quad (1.29)$$

où

$$\Phi_G (a) = \frac{e^{\text{ad}_G(a)} - 1}{\text{ad}_G(a)} = \sum_{n=0}^{\infty} \frac{1}{(n+1)!} \text{ad}_G(a)^n = Id_p + \frac{1}{2} \text{ad}_G(a) + \dots \quad (1.30)$$

est parfois appelée *matrice Jacobienne à gauche de G*.

En utilisant le fait que $\log_G^\vee (X^{-1}) = -\log_G^\vee (X)$, nous avons également la formule suivante :

$$\begin{aligned} \log_G^\vee ((\exp_G^\wedge (-a) \exp_G^\wedge (a+b))^{-1}) &= \log_G^\vee (\exp_G^\wedge (-a-b) \exp_G^\wedge (a)) \\ &= -\Phi_G (-a) b + O(\|b\|^2) \end{aligned} \quad (1.31)$$

Ainsi en définissant le changement de variable $c = -a$ et $d = -b$, nous obtenons la formule miroir de (1.29) :

$$\log_G^\vee (\exp_G^\wedge (c+d) \exp_G^\wedge (-c)) = \Phi_G (c) d + O(\|d\|^2) \quad (1.32)$$

1.1.4.2 Seconde formule BCH

D'après [Miller 1972], Théorème 5.5, nous obtenons le développement de Taylor à l'ordre 1 suivant :

$$\log_G^\vee (\exp_G^\wedge (a) \exp_G^\wedge (b)) = b + \varphi_G (b) a + O(\|a\|^2) \quad (1.33)$$

où

$$\varphi_G (b) = \frac{\text{ad}_G(b)}{e^{\text{ad}_G(b)} - 1} = \sum_{n=0}^{\infty} \frac{B_n \text{ad}_G(b)^n}{n!} = Id_p - \frac{1}{2} \text{ad}_G(b) + \dots \quad (1.34)$$

est l'inverse de la *matrice Jacobienne à gauche de G*, c'est à dire :

$$\Phi_G (a)^{-1} = \varphi_G (a) \quad (1.35)$$

Les coefficients B_n sont les nombres de Bernoulli [Ireland 1990].

1.1.5 Dérivées première et seconde de l'application logarithme de matrice

Nous nous intéressons ici aux expressions des dérivées première et seconde de l'application logarithme de matrice qui nous seront utiles pour le chapitre suivant. A notre connaissance, c'est la première fois que cette expression de la dérivée seconde est obtenue.

1.1.5.1 Dérivées première et seconde le long d'une courbe

Rappelons d'abord la définition du logarithme de matrice :

$$\log_G (Id + A) = \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k} A^k \quad (1.36)$$

$$= A - \frac{A^2}{2} + \frac{A^3}{3} + O(\|A\|^4) \quad (1.37)$$

où $A \in \mathbb{R}^{n \times n}$ et où nous supposons que la série entière de $\log_G (Id + A)$ converge. Les dérivées première et seconde le long d'une courbe peuvent alors être obtenues de la manière suivante :

$$\begin{aligned} \frac{d}{ds} \log_G (Id + A(s)) &= \frac{dA(s)}{ds} - \frac{1}{2} \left(A(s) \frac{dA(s)}{ds} + \frac{dA(s)}{ds} A(s) \right) \\ &+ \frac{1}{3} \left(\frac{dA(s)}{ds} A(s)^2 + A(s) \frac{dA(s)}{ds} A(s) + A(s)^2 \frac{dA(s)}{ds} \right) + O(\|A(s)\|^3) \frac{dA(s)}{ds} \end{aligned} \quad (1.38)$$

En partant de l'expression précédente nous avons alors :

$$\begin{aligned} \frac{d^2}{ds^2} \log_G (Id + A(s)) &= \frac{d^2 A(s)}{ds^2} - \frac{1}{2} \left(A(s) \frac{d^2 A(s)}{ds^2} + \frac{d^2 A(s)}{ds^2} A(s) \right) - \left(\frac{dA(s)}{ds} \right)^2 \\ &+ \frac{2}{3} \left(\left(\frac{dA(s)}{ds} \right)^2 A(s) + \frac{dA(s)}{ds} A(s) \frac{dA(s)}{ds} + A(s) \left(\frac{dA(s)}{ds} \right)^2 \right) \\ &+ O(\|A(s)\|^2) \left(\frac{dA(s)}{ds}, \frac{dA(s)}{ds} \right) + O(\|A(s)\|^3) \frac{d^2 A(s)}{ds^2} \end{aligned} \quad (1.39)$$

où la notation $O(\|\cdot\|)(\cdot, \cdot)$ est définie de la manière suivante :

$O(\|a\|^k)(b, c) = F(a)(b, c)$ où $F(a)(b, c)$ est une application bilinéaire en b et c et où il existe une constante $C > 0$ tel que $\sup_{b, c \text{ avec } \|b\|=1, \|c\|=1} \|F(a)(b, c)\| \leq C \|a\|^k$.

1.1.5.2 Dérivée première de l'application logarithme de matrice

Définissons tout d'abord :

$$Id + A(s) = Y \exp_G(s\mathfrak{c}) \quad (1.40)$$

où $Y \in G$ et $\mathfrak{c} \in \mathfrak{g}$. Alors :

$$\frac{dA(s)}{ds} = Y \exp_G(s\mathfrak{c}) \mathfrak{c} \quad (1.41)$$

$$\frac{d^2 A(s)}{ds^2} = Y \exp_G(s\mathfrak{c}) \mathfrak{c}^2 \quad (1.42)$$

$$\left. \frac{dA(s)}{ds} \right|_{s=0} = Y \mathfrak{c} \quad (1.43)$$

$$\left. \frac{d^2 A(s)}{ds^2} \right|_{s=0} = Y \mathfrak{c}^2 \quad (1.44)$$

Ainsi, (1.38) s'exprime alors de la forme suivante :

$$\left. \frac{d}{ds} \log_G(Y \exp_G(s\mathfrak{c})) \right|_{s=0} \quad (1.45)$$

$$\begin{aligned} &= Y \mathfrak{c} - \frac{1}{2} ((Y - Id) Y \mathfrak{c} + Y \mathfrak{c} (Y - Id)) \\ &+ \frac{1}{3} (Y \mathfrak{c} (Y - Id)^2 + (Y - Id) Y \mathfrak{c} (Y - Id) + (Y - Id)^2 Y \mathfrak{c}) + O(\|(Y - Id)\|^3) Y \mathfrak{c} \end{aligned} \quad (1.46)$$

En définissant $Y = \exp_G(\mathfrak{b})$ et en utilisant les résultats de l'annexe B, nous obtenons finalement :

$$\begin{aligned} &\left. \frac{d}{ds} \log_G(\exp_G(\mathfrak{b}) \exp_G(s\mathfrak{c})) \right|_{s=0} \\ &= \left(Id + \mathfrak{b} + \frac{1}{2} \mathfrak{b}^2 \right) \mathfrak{c} - \frac{1}{2} \left(\left(\mathfrak{b} + \frac{3}{2} \mathfrak{b}^2 \right) \mathfrak{c} + \mathfrak{c} \left(\mathfrak{b} + \frac{1}{2} \mathfrak{b}^2 \right) + \mathfrak{b} \mathfrak{c} \mathfrak{b} \right) \\ &+ \frac{1}{3} (\mathfrak{c} \mathfrak{b}^2 + \mathfrak{b} \mathfrak{c} \mathfrak{b} + \mathfrak{b}^2 \mathfrak{c}) + O(\|\mathfrak{b}\|^3) \mathfrak{c} \end{aligned} \quad (1.47)$$

$$= \mathfrak{c} + \frac{1}{2} \mathfrak{b} \mathfrak{c} - \frac{1}{2} \mathfrak{c} \mathfrak{b} + \frac{1}{12} \mathfrak{c} \mathfrak{b}^2 + \frac{1}{12} \mathfrak{b}^2 \mathfrak{c} - \frac{2}{12} \mathfrak{b} \mathfrak{c} \mathfrak{b} + O(\|\mathfrak{b}\|^3) \mathfrak{c} \quad (1.48)$$

$$= \mathfrak{c} + \frac{1}{2} [\mathfrak{b}, \mathfrak{c}] + \frac{1}{12} [\mathfrak{b}, [\mathfrak{b}, \mathfrak{c}]] + O(\|\mathfrak{b}\|^3) \mathfrak{c} \quad (1.49)$$

où nous avons utilisé la propriété suivante du crochet de Lie ($\mathfrak{a}, \mathfrak{b} \in \mathfrak{g}$) :

$$[\mathfrak{a}, [\mathfrak{a}, \mathfrak{b}]] = \mathfrak{a} \mathfrak{a} \mathfrak{b} + \mathfrak{b} \mathfrak{a} \mathfrak{a} - 2 \mathfrak{a} \mathfrak{b} \mathfrak{a} \quad (1.50)$$

1.1.5.3 Dérivée seconde de l'application logarithme de matrice

D'après (1.39) et en utilisant (1.40), nous obtenons l'expression suivante :

$$\left. \frac{d^2}{ds^2} \log_G (Y \exp_G (s\mathfrak{c})) \right|_{s=0} \quad (1.51)$$

$$\begin{aligned} &= Y\mathfrak{c}^2 - \frac{1}{2} \left((Y - Id) Y \mathfrak{c}^2 + Y \mathfrak{c}^2 (Y - Id) \right) - (Y\mathfrak{c})^2 \\ &+ \frac{2}{3} \left((Y\mathfrak{c})^2 (Y - Id) + Y\mathfrak{c} (Y - Id) Y\mathfrak{c} + (Y - Id) (Y\mathfrak{c})^2 \right) + O \left(\|(Y - Id)\|^2 \right) (\mathfrak{c}, \mathfrak{c}) \end{aligned} \quad (1.52)$$

En définissant $Y = \exp_G (\mathfrak{b})$ et en utilisant les résultats de l'annexe B, nous obtenons finalement :

$$\left. \frac{d^2}{ds^2} \log_G (\exp_G (\mathfrak{b}) \exp_G (s\mathfrak{c})) \right|_{s=0} \quad (1.53)$$

$$\begin{aligned} &= (Id + \mathfrak{b}) \mathfrak{c}^2 - \frac{1}{2} \left(\mathfrak{b}\mathfrak{c}^2 + \mathfrak{c}^2 \mathfrak{b} \right) - \left(\mathfrak{c}^2 + \mathfrak{c}\mathfrak{b}\mathfrak{c} + \mathfrak{b}\mathfrak{c}^2 \right) \\ &+ \frac{2}{3} \left(\mathfrak{c}^2 \mathfrak{b} + \mathfrak{c}\mathfrak{b}\mathfrak{c} + \mathfrak{b}\mathfrak{c}^2 \right) + O \left(\|\mathfrak{b}\|^2 \right) (\mathfrak{c}, \mathfrak{c}) \end{aligned} \quad (1.54)$$

$$= \frac{1}{6} \mathfrak{b}\mathfrak{c}^2 + \frac{1}{6} \mathfrak{c}^2 \mathfrak{b} - \frac{2}{6} \mathfrak{c}\mathfrak{b}\mathfrak{c} + O \left(\|\mathfrak{b}\|^2 \right) (\mathfrak{c}, \mathfrak{c}) \quad (1.55)$$

$$= \frac{1}{6} [\mathfrak{c}, [\mathfrak{c}, \mathfrak{b}]] + O \left(\|\mathfrak{b}\|^2 \right) (\mathfrak{c}, \mathfrak{c}) \quad (1.56)$$

1.1.6 Exemples de groupes de Lie

Il existe de nombreux groupes de Lie (matriciels). Nous donnons ici quelques exemples qui sont utilisés par la suite dans les différentes applications que nous considérons.

1.1.6.1 Espace euclidien : \mathbb{R}^n

L'application suivante permet de transformer l'espace euclidien \mathbb{R}^p en un groupe de Lie matriciel commutatif :

$$x \in \mathbb{R}^p \rightarrow X = \begin{bmatrix} Id_p & x \\ 0 & 1 \end{bmatrix} \subset \mathbb{R}^{(p+1) \times (p+1)} \quad (1.57)$$

La multiplication matricielle correspond bien à l'addition :

$$X_1 X_2 = \begin{bmatrix} Id_p & x_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} Id_p & x_2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} Id_p & x_1 + x_2 \\ 0 & 1 \end{bmatrix} = X_2 X_1 \quad (1.58)$$

Et les applications \exp_G et \log_G sont triviales :

$$\log_{\mathbb{R}^n} \left(\begin{bmatrix} Id_p & x \\ 0 & 1 \end{bmatrix} \right) = \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k} \left(\begin{bmatrix} Id_p & x \\ 0 & 1 \end{bmatrix} - Id_{p+1} \right)^k \quad (1.59)$$

$$= \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k} \begin{bmatrix} 0 & x \\ 0 & 0 \end{bmatrix}^k = \begin{bmatrix} 0 & x \\ 0 & 0 \end{bmatrix} \quad (1.60)$$

$$\exp_{\mathbb{R}^n} \left(\begin{bmatrix} 0 & x \\ 0 & 0 \end{bmatrix} \right) = \sum_{k=0}^{\infty} \frac{1}{k!} \begin{bmatrix} 0 & x \\ 0 & 0 \end{bmatrix}^k = Id_{p+1} + \begin{bmatrix} 0 & x \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} Id_p & x \\ 0 & 1 \end{bmatrix} \quad (1.61)$$

Un espace euclidien est donc un groupe de Lie trivial, et c'est en ce sens que la plupart des algorithmes proposés dans ce manuscrit généralisent les algorithmes destinés à l'estimation de paramètres euclidiens.

1.1.6.2 Groupe multiplicatif : (\mathbb{R}^+*, \times)

Le groupe multiplicatif, défini par les éléments de \mathbb{R}^+* munis de la multiplication scalaire, est un groupe de Lie commutatif permettant en pratique d'estimer des facteurs d'échelle. Ce groupe de Lie étant de dimension 1, ses applications \exp_G et \log_G correspondent aux fonctions exponentielle et logarithme d'un scalaire. Plus loin dans ce chapitre nous verrons le lien entre ce groupe de Lie et la distribution de probabilité Log-Normale.

1.1.6.3 Groupe Spécial Orthogonal en dimension 2 : $SO(2)$

Le groupe de Lie $SO(2)$ intervient dès lors que l'on souhaite estimer une orientation dans un plan. Il est défini comme :

$$SO(2) = \left\{ R \in GL(2) \left| R^T R = Id, \det(R) = 1 \right. \right\} \quad (1.62)$$

Pour trouver la forme de son algèbre de Lie, nous pouvons par exemple prendre la dérivée d'une courbe $R(t) \in SO(2)$ passant par l'identité ($R(t=0) = Id$) et exploiter la contrainte d'orthonormalité :

$$\frac{d}{dt} R^T R = \frac{dR^T}{dt} R + R^T \frac{dR}{dt} = 0 \quad (1.63)$$

En $t = 0$ nous avons $R(t=0) = Id$ et donc :

$$\frac{dR(t=0)^T}{dt} + \frac{dR(t=0)}{dt} = 0 \quad (1.64)$$

$\frac{dR(t=0)}{dt}$ est par conséquent une matrice antisymétrique. Or $\frac{dR(t=0)}{dt}$ est également un élément de l'algèbre de Lie $\mathfrak{so}(2)$ de $SO(2)$. Nous pouvons donc représenter $\mathfrak{so}(2)$ de la manière suivante :

$$\mathfrak{so}(2) = \left\{ \theta \in \mathbb{R} \left| [\theta]_{SO(2)}^\wedge = \begin{bmatrix} 0 & -\theta \\ \theta & 0 \end{bmatrix} \right. \right\} \quad (1.65)$$

Chaque élément de ce groupe représente une matrice de rotation en dimension 2 et θ correspond à l'angle de rotation. Ce groupe de Lie de dimension 1 est commutatif.

1.1.6.4 Groupe Spécial Orthogonal en dimension 3 : $SO(3)$

Le groupe de Lie $SO(3)$ intervient notamment dans les problèmes où il faut estimer l'orientation d'un objet tridimensionnel. Il est défini comme :

$$SO(3) = \left\{ R \in GL(3) \left| R^T R = Id, \det(R) = 1 \right. \right\} \quad (1.66)$$

Son algèbre de Lie $\mathfrak{so}(3)$ peut être obtenu comme nous l'avons fait pour $\mathfrak{so}(2)$. Nous pouvons également employer une autre manière qui utilise, elle aussi, la contrainte d'orthonormalité :

$$\exp_{SO(3)} \left([\omega]_{SO(3)}^\wedge \right)^T = \exp_{SO(3)} \left([\omega]_{SO(3)}^\wedge \right)^T \quad (1.67)$$

$$= \exp_{SO(3)} \left([\omega]_{SO(3)}^\wedge \right)^{-1} = \exp_{SO(3)} \left(-[\omega]_{SO(3)}^\wedge \right) \quad (1.68)$$

$[\omega]_{SO(3)}^\wedge$ qui est un élément de $\mathfrak{so}(3)$ est donc une matrice antisymétrique. Nous pouvons alors représenter $\mathfrak{so}(3)$ de la manière suivante [Selig 2005] :

$$\mathfrak{so}(3) = \left\{ \omega \in \mathbb{R}^3 \left| [\omega]_{SO(3)}^\wedge = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_{zy} & \omega_x & 0 \end{bmatrix} \right. \right\} \quad (1.69)$$

Chaque élément de ce groupe représente une matrice de rotation en dimension 3. La direction de ω correspond à l'axe de rotation et sa longueur indique l'angle.

1.1.6.5 Groupe Spécial Euclidien en dimension 2 : $SE(2)$

Le groupe de Lie $SE(2)$ est en général utilisé pour modéliser le déplacement planaire d'un robot qui a deux degrés de liberté en translation et un en rotation. Ce groupe de Lie correspond au produit semi-direct $SO(2) \ltimes \mathbb{R}^2$ et est défini

comme :

$$SE(2) = \left\{ C = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \middle| R \in SO(2), t \in \mathbb{R}^2 \right\} \quad (1.70)$$

Nous pouvons représenter son algèbre de Lie $\mathfrak{se}(2)$ de la manière suivante [Long 2012] :

$$\mathfrak{se}(2) = \left\{ \begin{bmatrix} \theta \\ u \end{bmatrix} \in \mathbb{R}^3 \middle| \left[\begin{bmatrix} \theta \\ u \end{bmatrix} \right]_{SE(2)}^\wedge = \begin{bmatrix} 0 & -\theta & u_x \\ \theta & 0 & u_y \\ 0 & 0 & 0 \end{bmatrix} \right\} \quad (1.71)$$

1.1.6.6 Groupe Spécial Euclidien en dimension 3 : $SE(3)$

Le groupe de Lie $SE(3)$ est en général utilisé pour modéliser le déplacement d'un objet tridimensionnel, comme une caméra par exemple, qui a trois degrés de liberté en translation et 3 en rotation. Ce groupe de Lie correspond au produit semi-direct (noté \ltimes) $SO(3) \ltimes \mathbb{R}^3$ et est défini comme :

$$SE(3) = \left\{ C = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \middle| R \in SO(3), t \in \mathbb{R}^3 \right\} \quad (1.72)$$

Nous pouvons représenter son algèbre de Lie $\mathfrak{se}(3)$ de la manière suivante [Selig 2005] :

$$\mathfrak{se}(3) = \left\{ \begin{bmatrix} \omega \\ u \end{bmatrix} \in \mathbb{R}^6 \middle| \left[\begin{bmatrix} \theta \\ u \end{bmatrix} \right]_{SE(3)}^\wedge = \begin{bmatrix} 0 & -\omega_z & \omega_y & u_x \\ \omega_z & 0 & -\omega_x & u_y \\ -\omega_y & \omega_x & 0 & u_z \\ 0 & 0 & 0 & 0 \end{bmatrix} \right\} \quad (1.73)$$

1.1.6.7 Groupe de Lie des Similitudes en dimension 3 : $Sim(3)$

Le groupe de Lie $Sim(3) = ((\mathbb{R}^+*, \times) \times SO(3)) \ltimes \mathbb{R}^3$ est notamment utilisé pour aligner deux nuages de points 3D dont l'orientation $SO(3)$, la position \mathbb{R}^3 et l'échelle (\mathbb{R}^+*, \times) diffèrent. Il est défini comme :

$$Sim(3) = \left\{ S = \begin{bmatrix} sR & t \\ 0 & 1 \end{bmatrix} \middle| R \in SO(3), t \in \mathbb{R}^3, s \in (\mathbb{R}^+*, \times) \right\} \quad (1.74)$$

Nous pouvons représenter son algèbre de Lie $\mathfrak{sim}(3)$ de la manière suivante [Strasdat 2012] :

$$\mathfrak{sim}(3) = \left\{ \begin{bmatrix} \lambda \\ \omega \\ u \end{bmatrix} \in \mathbb{R}^7 \mid \left[\begin{bmatrix} \lambda \\ \omega \\ u \end{bmatrix} \right]_{Sim(3)}^\wedge = \begin{bmatrix} \lambda & -\omega_z & \omega_y & u_x \\ \omega_z & \lambda & -\omega_x & u_y \\ -\omega_y & \omega_x & \lambda & u_z \\ 0 & 0 & 0 & 0 \end{bmatrix} \right\} \quad (1.75)$$

1.1.6.8 Groupe Spécial Linéaire : $SL(3)$

Le groupe de Lie $SL(3)$ est en général utilisé pour représenter une homographie en vision par ordinateur. Il est défini comme :

$$SL(3) = \left\{ H \in GL(3) \mid \det(H) = 1 \right\} \quad (1.76)$$

La matrice H a 9 éléments et une seule contrainte ($\det(H) = 1$). Par conséquent, $SL(3)$ est un groupe de Lie de dimension 8. Son algèbre de Lie doit respecter la propriété suivante :

$$\det(\exp_{SL(3)}(\mathfrak{h})) = e^{\text{tr}(\mathfrak{h})} = 1 \quad (1.77)$$

Donc

$$\text{tr}(\mathfrak{h}) = 0 \quad (1.78)$$

Nous pouvons représenter l'algèbre de Lie $\mathfrak{sl}(3)$ de $SL(3)$ de la manière suivante [Benhimane 2007] :

$$\mathfrak{sl}(3) = \left\{ h \in \mathbb{R}^8 \mid [h]_{SL(3)}^\wedge = \begin{bmatrix} h_5 & h_3 & h_1 \\ h_4 & -h_5 - h_6 & h_2 \\ h_7 & h_8 & h_6 \end{bmatrix} \right\} \quad (1.79)$$

1.1.7 Transformation géométrique et interpolation

1.1.7.1 Transformation géométrique

Les groupes de Lie matriciels considérés paragraphe 1.1.6 peuvent en réalité tous être interprétés comme des transformations entre deux référentiels i et j . Cette représentation nous permet notamment de ne plus simplement manipuler des concepts abstraits mais des objets géométriques.

Prenons par exemple le cas de $SE(3)$. Soit $x^j = \overrightarrow{O_j}x^j = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \in \mathbb{R}^3$ un

vecteur défini dans le référentiel j qui a pour base $\left(O_j; \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right)$.

En coordonnées homogènes x^j s'écrit $\underline{x}^j = \begin{bmatrix} \overrightarrow{O_j x^j} \\ 1 \end{bmatrix}$. Le fait d'appliquer la matrice $C_{ij} = \begin{bmatrix} R_{ij} & t_{ij} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{ij} & \overrightarrow{O_i O_j^i} \\ 0 & 1 \end{bmatrix} \in SE(3)$ à \underline{x}^j permet d'effectuer une rotation suivie d'une translation du point x afin de l'exprimer dans le référentiel i :

$$C_{ij}\underline{x}^j = \begin{bmatrix} R_{ij} & \overrightarrow{O_i O_j^i} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \overrightarrow{O_j x^j} \\ 1 \end{bmatrix} = \begin{bmatrix} R_{ij}\overrightarrow{O_j x^j} + \overrightarrow{O_i O_j^i} \\ 1 \end{bmatrix} \quad (1.80)$$

$$= \begin{bmatrix} \overrightarrow{O_j x^i} + \overrightarrow{O_i O_j^i} \\ 1 \end{bmatrix} = \begin{bmatrix} \overrightarrow{O_i x^i} \\ 1 \end{bmatrix} = \underline{x}^i \quad (1.81)$$

Ainsi C_{ij} permet de passer du référentiel j au référentiel i . La multiplication matricielle reste cohérente :

$$C_{ij}C_{jk} = \begin{bmatrix} R_{ij} & \overrightarrow{O_i O_j^i} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{jk} & \overrightarrow{O_j O_k^j} \\ 0 & 1 \end{bmatrix} \quad (1.82)$$

$$= \begin{bmatrix} R_{ij}R_{jk} & R_{ij}\overrightarrow{O_j O_k^j} + \overrightarrow{O_i O_j^i} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{ik} & \overrightarrow{O_i O_k^i} \\ 0 & 1 \end{bmatrix} = C_{ik} \quad (1.83)$$

Tout comme l'inverse matriciel :

$$C_{ji}C_{ij} = \begin{bmatrix} R_{ji} & \overrightarrow{O_j O_i^j} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{ij} & \overrightarrow{O_i O_j^i} \\ 0 & 1 \end{bmatrix} \quad (1.84)$$

$$= \begin{bmatrix} R_{ji}R_{ij} & R_{ji}\overrightarrow{O_i O_j^i} + \overrightarrow{O_j O_i^j} \\ 0 & 1 \end{bmatrix} = Id \quad (1.85)$$

Un élément $\delta_{jj'} \in \mathbb{R}^6$ de l'algèbre de Lie $\mathfrak{se}(3)$ peut alors être associé à une transformation entre un référentiel j' et un référentiel j grâce à l'application exponentielle :

$$C_{jj'} = \exp_{SE(3)}^{\wedge}(\delta_{jj'}) \quad (1.86)$$

La notion de représentation adjointe de $SE(3)$ sur $\mathfrak{se}(3)$ qui traduit la non commutativité du groupe peut également être interprétée géométriquement :

$$C_{ij}\exp_{SE(3)}^{\wedge}(\delta_{jj'}) = \exp_{SE(3)}^{\wedge}(\text{Ad}_{SE(3)}(C_{ij})\delta_{jj'})C_{ij} = \exp_{SE(3)}^{\wedge}(\delta_{i'i})C_{ij} \quad (1.87)$$

Cela signifie que l'action d'un incrément $\delta_{jj'}$ par la droite n'est pas équivalent

à l'action de cet incrément par la gauche. Pour obtenir le même résultat, en utilisant l'action à gauche, il faut utiliser l'opérateur $\text{Ad}_{SE(3)}(\cdot)$.

1.1.7.2 Interpolation

Un problème intéressant, permettant d'exploiter les principales propriétés des groupes de Lie matriciels et ainsi de mieux se les approprier, est de considérer l'interpolation sur groupe de Lie. Une méthode intuitive pour interpoler entre deux transformations C_{ij} et C_{ik} est de considérer l'erreur $\log_{SE(3)}^\vee(C_{ij}C_{ik}^{-1})$. En faisant varier une variable t entre 0 et 1 nous obtenons alors :

$$C_1(t) = C_{ij} \exp_{SE(3)}^\wedge \left(t \log_{SE(3)}^\vee (C_{ij}^{-1} C_{ik}) \right) \quad (1.88)$$

En $t = 0$, nous avons bien :

$$C_1(t = 0) = C_{ij} \quad (1.89)$$

et en $t = 1$, nous avons bien :

$$C_1(t = 1) = C_{ij} \exp_{SE(3)}^\wedge \left(\log_{SE(3)}^\vee (C_{ij}^{-1} C_{ik}) \right) = C_{ik} \quad (1.90)$$

Nous avons donc défini une méthode pour interpoler de j vers k en utilisant des incréments agissant à droite de C_{ij} . On peut maintenant se poser la même question mais en utilisant des incréments qui agissent à gauche de C_{ij} . Nous obtenons alors :

$$C_2(t) = \exp_{SE(3)}^\wedge \left(t \log_{SE(3)}^\vee (C_{ik} C_{ij}^{-1}) \right) C_{ij} \quad (1.91)$$

où $C_2(t = 0) = C_{ij}$ et $C_2(t = 1) = C_{ik}$.

Normalement, dans une interpolation, l'utilisation d'incrémentes à droite ou à gauche devrait produire le même résultat. Nous nous proposons de montrer que c'est ici le cas, c'est à dire que $C_2(t) = C_1(t)$.

$$C_2(t) = \exp_{SE(3)}^\wedge \left(t_1 \log_{SE(3)}^\vee (C_{ik} C_{ij}^{-1}) \right) C_{ij} \quad (1.92)$$

$$= C_{ij} \exp_{SE(3)}^\wedge \left(t_1 \text{Ad}_{SE(3)} \left(C_{ij}^{-1} \right) \log_{SE(3)}^\vee (C_{ik} C_{ij}^{-1}) \right) \quad (1.93)$$

Nous devons donc prouver que :

$$\log_{SE(3)}^\vee (C_{ij}^{-1} C_{ik}) = \text{Ad}_{SE(3)} \left(C_{ij}^{-1} \right) \log_{SE(3)}^\vee (C_{ik} C_{ij}^{-1}) \quad (1.94)$$

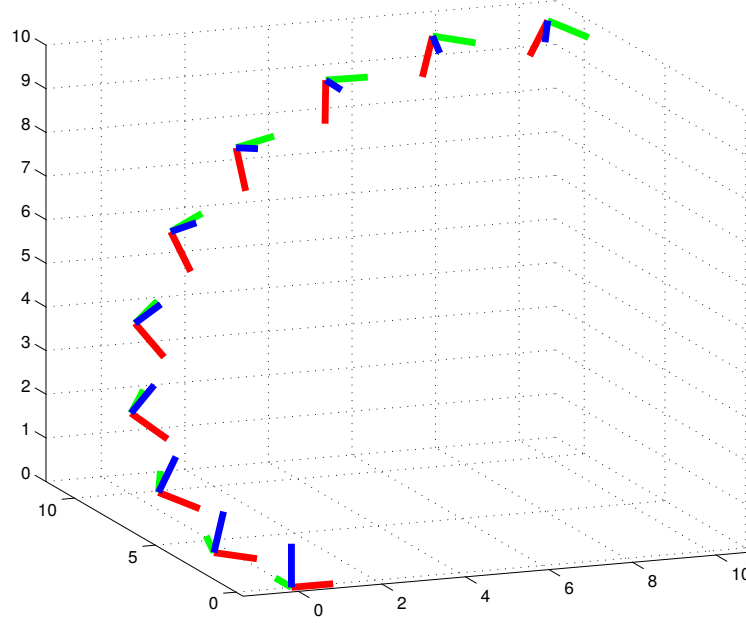


Figure 1.4 – Résultat d’une interpolation sur $SE(3)$ où $C(t=0) = Id$ et $C(t=1) = \begin{bmatrix} \exp_{SE(3)}^{\wedge} \left(\begin{bmatrix} \frac{\pi}{2} & \frac{\pi}{2} & -\frac{\pi}{2} \end{bmatrix}^T \right) & \begin{bmatrix} 10 & 10 & 10 \end{bmatrix}^T \\ 0 & 1 \end{bmatrix}$

Et en effet :

$$\exp_{SE(3)}^{\wedge} \left(\text{Ad}_{SE(3)} \left(C_{ij}^{-1} \right) \log_{SE(3)}^{\vee} \left(C_{ik} C_{ij}^{-1} \right) \right) \quad (1.95)$$

$$= C_{ij}^{-1} \exp_{SE(3)}^{\wedge} \left(\log_{SE(3)}^{\vee} \left(C_{ik} C_{ij}^{-1} \right) \right) C_{ij} = C_{ij}^{-1} C_{ik} C_{ij}^{-1} C_{ij} = C_{ij}^{-1} C_{ik} \quad (1.96)$$

Un exemple d’interpolation sur $SE(3)$ est proposé en figure 1.4.

1.1.8 Détails d’implémentation

1.1.8.1 Produit de groupes de Lie

Pour les différentes applications que nous considérons dans ce manuscrit, nous souhaitons proposer des algorithmes permettant d’estimer conjointement plusieurs paramètres évoluant sur des groupes de Lie matriciels.

La manière la plus simple de “concaténer” plusieurs paramètres évoluant sur des groupes de Lie est de considérer le groupe de Lie formé par leur produit direct. Par exemple, si $X \in G$ et $Y \in G'$ alors $\begin{bmatrix} X & 0 \\ 0 & Y \end{bmatrix} \in G \times G'$ où G et G' sont deux groupes de Lie matriciels. Remarquons qu’il existe d’autres manières de former des groupes de Lie [Rudkovskii 1997], comme le produit semi-direct ou encore le produit “twisted”. Dans ce manuscrit, nous utilisons exclusivement le produit direct de groupes de Lie existant déjà dans la littérature (voir

paragraphe 1.1.6).

Puisque le *produit* (direct, semi-direct, etc.) de groupes de Lie est lui-même un groupe de Lie, un algorithme développé pour estimer un paramètre évoluant sur un groupe de Lie matriciel *abstrait* pourra être utilisé pour estimer plusieurs paramètres conjointement en “concaténant” ces derniers à l’aide de l’un des produits.

1.1.8.2 Opérateurs ad_G et Ad_G

Pour un groupe de Lie matriciel G , une fois que nous avons défini les fonctions $[\cdot]_G^\wedge$ et $[\cdot]_G^\vee$, l’expression des opérateurs ad_G et Ad_G peut être directement déduite en utilisant leurs définitions respectives (1.20) et (1.25).

1.1.8.3 Applications \exp_G^\wedge et \log_G^\vee

De la même manière que pour ad_G et Ad_G , une fois que nous avons défini les fonctions $[\cdot]_G^\wedge$ et $[\cdot]_G^\vee$, il suffit d’utiliser les fonctions exponentielle et logarithme d’une matrice qui sont déjà pré-codées dans de nombreuses bibliothèques ainsi qu’en Matlab pour pouvoir calculer \exp_G^\wedge et \log_G^\vee . Des informations sur comment calculer efficacement ces deux fonctions sont disponibles dans [Higham 2008].

Pour certains groupes de Lie comme $SO(2)$ [Long 2012], $SE(2)$ [Long 2012], $SO(3)$ [Selig 2005], $SE(3)$ [Selig 2005] ou encore $Sim(3)$ [Strasdat 2012], il existe des formules analytiques de \exp_G^\wedge et \log_G^\vee qui permettent de respecter plus facilement les contraintes de temps réel requises dans certaines applications. Cependant pour $SL(3)$ par exemple, à notre connaissance, il n’existe pas de formule analytique.

1.1.8.4 Jacobienne à gauche du groupe de Lie $\Phi_G(\cdot)$ et son inverse $\varphi_G(\cdot)$

Les matrices $\Phi_G(\cdot)$ et $\varphi_G(\cdot)$ sont définies par des séries entières faisant intervenir l’opérateur ad_G . De la même manière que pour \exp_G^\wedge et \log_G^\vee , il existe des formules analytiques pour certains groupes de Lie comme $SO(2)$, $SE(2)$, $SO(3)$, $SE(3)$ ou encore $Sim(3)$ (voir [Barfoot 2014]). Cependant, pour $SL(3)$ par exemple, à notre connaissance, il n’existe pas de formule analytique. En pratique, que ce soit pour $\Phi_G(\cdot)$ ou $\varphi_G(\cdot)$, nous tronquons la série entière après les premiers termes.

1.2 La distribution normale concentrée sur groupe de Lie matriciel

Afin de pouvoir développer des méthodes d'inférence bayésienne pour l'estimation de paramètres évoluant sur des groupes de Lie matriciels, nous devons disposer de distributions de probabilité définies sur les groupes de Lie. Dans ce manuscrit nous nous intéressons à une distribution appelée *distribution normale concentrée sur groupe de Lie* qui possède une interprétation géométrique intuitive, basée sur une distribution gaussienne dans l'algèbre de Lie. De plus, cette distribution a l'avantage d'être définie pour tout groupe de Lie (matriciel). Les algorithmes développés pour une telle distribution pourront donc s'appliquer à tous les groupes de Lie présentés paragraphe 1.1.6.

1.2.1 Définition

Nous introduisons ici le concept de distribution *normale concentrée sur groupe de Lie*, qui est une généralisation possible de la distribution normale, définie pour des paramètres euclidiens, aux groupes de Lie. Cette distribution a déjà été employée dans différents ouvrages, tels que [Smith 2003; Barfoot 2014], portant sur l'estimation de paramètres sur groupe de Lie.

Une variable aléatoire $X \in G$ suit une distribution *normale concentrée sur groupe de Lie* à gauche de “moyenne” μ et de “covariance” P , que l'on note $X \sim \mathcal{N}_G^L(X; \mu, P)$, si :

$$X = \mu \exp_G^\wedge(\epsilon) \quad (1.97)$$

où $\epsilon \sim \mathcal{N}_{\mathbb{R}^p}(\epsilon; \mathbf{0}_{p \times 1}, P)$ est une distribution normale multivariée centrée de covariance P définie dans l'algèbre de Lie de G :

$$p(\epsilon) = \frac{1}{\sqrt{(2\pi)^p \det(P)}} e^{-\frac{1}{2} \|\epsilon\|_P^2} \quad (1.98)$$

où

$$\|\epsilon\|_P^2 = \epsilon^T P^{-1} \epsilon \quad (1.99)$$

est la distance de Mahalanobis. $P \subset \mathbb{R}^{p \times p}$ est par conséquent une matrice symétrique semi-définie positive. Cette distribution *normale concentrée sur groupe de Lie* à gauche revient donc à :

1. définir une distribution normale multivariée centrée de covariance P dans l'algèbre de Lie de G
2. transformer cette distribution en une distribution sur le groupe de Lie en utilisant l'application exponentielle

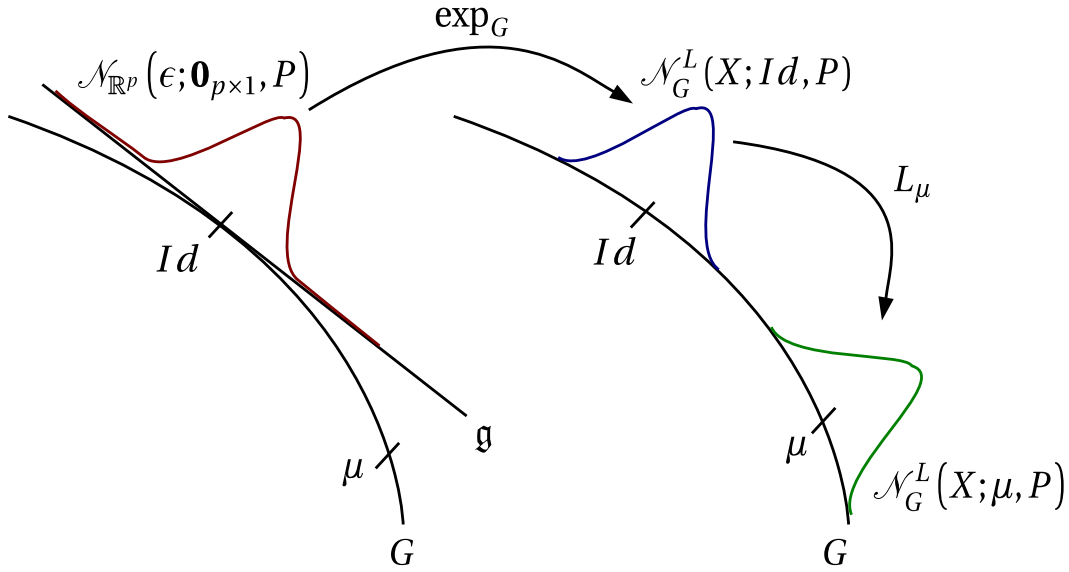


Figure 1.5 – Illustration graphique de la distribution *normale concentrée sur groupe de Lie à gauche*

3. transporter cette distribution autour de sa “moyenne” μ en utilisant l’action à gauche du groupe de Lie sur lui-même

Une illustration graphique de cette distribution est présentée en figure 1.5.

De la même manière, il est possible de définir une distribution *normale concentrée sur groupe de Lie à droite* de “moyenne” μ et de “covariance” P , que l’on note $X \sim \mathcal{N}_G^R(X; \mu, P)$:

$$X = \exp_G^\wedge(\epsilon) \mu \quad (1.100)$$

Le fait d’utiliser une distribution *normale concentrée sur groupe de Lie à gauche* (1.97) ou à droite (1.100) est un choix à effectuer qui dépend de l’application. Ce choix peut avoir une influence sur la simplicité des calculs.

En revanche, il est toujours possible de trouver une représentation équivalente d’une distribution normale concentrée à gauche sous la forme d’une distribution concentrée à droite en utilisant l’opérateur adjoint :

$$X = \mu \exp_G^\wedge(\epsilon) = \mu \exp_G^\wedge(\epsilon) \mu^{-1} \mu = \exp_G^\wedge(\text{Ad}_G(\mu) \epsilon) \mu \quad (1.101)$$

Ainsi la distribution $\mathcal{N}_G^L(X; \mu, P)$ est équivalente à la distribution $\mathcal{N}_G^R(X; \mu, \text{Ad}_G(\mu) P \text{Ad}_G(\mu)^T)$.

Dans la suite de ce manuscrit, nous utiliserons l’une ou l’autre en fonction du problème considéré. Nous présentons cependant ci-après uniquement la version à gauche, le raisonnement étant facilement reproductible pour la version à droite.

1.2.1.1 Expression de $p(X)$

La définition (1.97) de la distribution *normale concentrée sur groupe de Lie* à gauche permet de définir indirectement une densité de probabilité pour X par l'intermédiaire de $p(\epsilon)$. Dans un certain nombre d'applications l'expression de $p(X)$ n'est pas nécessaire (voir [Barfoot 2014]). Cependant, dans notre contexte, nous souhaitons définir des fonctions de vraisemblance ainsi que des a priori ayant la forme de distributions normales concentrées. Cela requiert l'expression explicite de $p(X)$. Afin d'obtenir cette expression nous devons d'abord introduire la notion de mesure de Haar invariante à gauche, que nous notons $d_H X$, qui respecte la propriété suivante ([Faraud 2008] Théorème 5.1.1) :

$$\int_G f(YX) d_H X = \int_G f(X) d_H X \quad (1.102)$$

pour tout $Y \in G$ où f est une fonction continue définie sur G . Cette mesure généralise la mesure de Lebesgue (qui est invariante par translation) et que l'on utilise habituellement pour définir des intégrales sur des espaces euclidiens. D'après la proposition 5.5.6 dans [Faraud 2008], si le support de f est contenu dans M (voir Fig.1.3) alors :

$$\int_G f(X) d_H X = c \int_{\mathbb{R}^p} f(\exp_G^\wedge(\epsilon)) \det(\Phi_G(-\epsilon)) d_L \epsilon \quad (1.103)$$

où $d_L \epsilon$ est la mesure de Lebesgue, $\Phi_G(\cdot)$ est la matrice Jacobienne à gauche de G (1.30) et c est une constante.

Afin de trouver l'expression de $p(X)$, nous pouvons donc utiliser (1.103) pour définir le changement de variable suivant :

$$\epsilon = \log_G^\vee(\mu^{-1}X) \quad (1.104)$$

ainsi que le changement de mesure suivant ² :

$$d_H X = \det(\Phi_G(-\epsilon)) d_L \epsilon \quad (1.105)$$

2. Remarquons que nous utilisons la mesure de Haar invariante à gauche puisque nous définissons ici la distribution normale concentrée à gauche. Pour la distribution normale concentrée à droite, nous utiliserions la mesure de Haar invariante à droite donnée par $d_H X = \det(\Phi_G(\epsilon)) d_L \epsilon$ (voir [Kirillov 1994] pp.141). Une manière de retrouver les expressions de ces deux mesures est de se rendre compte de leur lien avec les formules BCH présentées paragraphe 1.1.4.1.

Ce qui nous permet d'expliciter $p(X)$:

$$\begin{aligned}
 \int_{\mathbb{R}^p} p(\epsilon) d_L \epsilon &= \int_{\mathbb{R}^p} \frac{1}{\sqrt{(2\pi)^p \det(P)}} e^{-\frac{1}{2} \|\epsilon\|_P^2} d_L \epsilon \\
 &= \int_G \frac{1}{\sqrt{(2\pi)^p \det(P)}} e^{-\frac{1}{2} \|\log_G^\vee(\mu^{-1}X)\|_P^2} \frac{1}{\det(\Phi_G(-\log_G^\vee(\mu^{-1}X)))} d_H X \\
 &= \int_G p(X) d_H X
 \end{aligned} \tag{1.106}$$

où

$$p(X) = \frac{1}{\sqrt{(2\pi)^p \det(\Phi_G(-\log_G^\vee(\mu^{-1}X)) P \Phi_G(-\log_G^\vee(\mu^{-1}X))^T)}} e^{-\frac{1}{2} \|\log_G^\vee(\mu^{-1}X)\|_P^2} \tag{1.107}$$

Remarquons que le changement de variable que nous avons utilisé repose sur l'équation (1.103) qui est uniquement valide si le support de f est contenu dans M . Or nous avons appliqué ce changement de variable à une distribution normale $p(\epsilon)$ qui possède des queues infinies.

Le changement de variable que nous avons effectué est donc valide uniquement lorsque $p(\epsilon)$ est *concentrée*, c'est à dire lorsque la plus grande valeur propre de P est "suffisamment" petite pour que la masse de probabilité qui n'est pas dans S (voir Fig.1.3) soit négligeable. C'est pourquoi nous parlons de distribution *normale concentrée sur groupe de Lie*. Les paramètres μ et P de $p(X)$ sont appelés "moyenne" et "covariance" respectivement car μ correspond à la définition du barycentre sur groupe de Lie :

$$\int_G \log_G^\vee(\mu^{-1}X) p(X) d_H X = \int_G \epsilon p(\epsilon) d_L \epsilon = 0 \tag{1.108}$$

alors que P correspond lui à la définition de covariance sur groupe de Lie :

$$\int_G \log_G^\vee(\mu^{-1}X) \log_G^\vee(\mu^{-1}X)^T p(X) d_H X = \int_G \epsilon \epsilon^T p(\epsilon) d_L \epsilon = P \tag{1.109}$$

1.2.1.2 Approximation de $p(X)$

La définition de distribution *normale concentrée sur groupe de Lie* est, comme son nom l'indique, définie pour une matrice de covariance ayant de faibles valeurs propres. Dans ce contexte, il semble raisonnable de tronquer la série entière de $\Phi_G(\cdot)$ à l'ordre zéro dans $p(X)$ de manière à ce que le terme $\Phi_G(-\log_G^\vee(\mu^{-1}X))$ devienne la matrice identité. Nous obtenons donc

l'expression suivante :

$$p(X) \approx \frac{1}{\sqrt{(2\pi)^p \det(P)}} e^{-\frac{1}{2} \|\log_G^\vee(\mu^{-1}X)\|_P^2} \quad (1.110)$$

Cette expression approchée est très pratique puisque le terme se situant devant l'exponentielle devient alors une constante de normalisation. Ce terme n'interviendra donc pas lorsque, par exemple, nous chercherons le mode d'une distribution a posteriori où l'a priori sur X sera donné par une distribution *normale concentrée sur groupe de Lie*. Le fait de ne pas considérer l'approximation de $p(X)$ (1.110) mais plutôt (1.107) en tant que distribution a priori n'est pas traité dans ce manuscrit et est laissé en tant que perspective.

Il est intéressant de noter que d'autres travaux [Wang 2006 ; Wolfe 2011 ; Chirikjian 2014] sont partis d'une autre densité de probabilité de la forme :

$$\rho(x) = \alpha e^{-\frac{1}{2} \|\log_G^\vee(\mu^{-1}X)\|_P^2} \quad (1.111)$$

où α est une constante de normalisation. En faisant ensuite l'hypothèse d'une matrice de covariance P ayant de faibles valeurs propres, cette constante de normalisation est approchée par celle d'une distribution normale multivariée, c'est à dire :

$$\alpha \approx \frac{1}{\sqrt{(2\pi)^p \det(P)}} \quad (1.112)$$

La distribution obtenue est alors la même que celle que nous considérons.

1.2.2 Reparamétrisation de la covariance

Nous cherchons ici à obtenir la forme de la distribution qu'on obtient dans l'algèbre de Lie lorsqu'on cherche à *déplier* une distribution normale concentrée $\mathcal{N}_G^R(X; \mu_b, P_b)$ autour d'un point μ_a qui n'est pas sa moyenne.

1.2.2.1 Déploiement d'une distribution normale concentrée

En effet, lorsqu'on *déplie* une distribution normale concentrée $\mathcal{N}_G^R(X; \mu_b, P_b)$ autour de sa moyenne μ_b , nous avons :

$$\log_G^\vee(X\mu_b^{-1}) = \log_G^\vee(\exp_G^\wedge(\epsilon_b) \mu_b \mu_b^{-1}) \approx \epsilon_b \quad (1.113)$$

Nous obtenons ainsi, par définition de la distribution normale concentrée, une distribution normale multivariée $\mathcal{N}_{\mathbb{R}^p}(\epsilon_b; \mathbf{0}_{p \times 1}, P_b)$.

En revanche, lorsqu'on *déplie* une distribution normale concentrée $\mathcal{N}_G^R(X; \mu_b, P_b)$ autour d'un point $\mu_a \neq \mu_b$, nous avons alors (en utilisant la

formule BCH (1.33)) :

$$\begin{aligned}\epsilon &= \log_G^\vee(X\mu_a^{-1}) \\ &= \log_G^\vee(\exp_G^\wedge(\epsilon_b)\mu_b\mu_a^{-1}) \\ &\approx \log_G^\vee(\mu_b\mu_a^{-1}) + \varphi_G(\log_G^\vee(\mu_b\mu_a^{-1}))\epsilon_b\end{aligned}\quad (1.114)$$

Ainsi, nous obtenons une distribution normale multivariée de la forme $\mathcal{N}_{\mathbb{R}^p}(\epsilon; \log_G^\vee(\mu_b\mu_a^{-1}), \varphi_G(\log_G^\vee(\mu_b\mu_a^{-1}))P_b\varphi_G(\log_G^\vee(\mu_b\mu_a^{-1}))^T)$.

Le fait de *déplier* une distribution normale concentrée autour d'un autre point que sa moyenne a donc pour effet logique de translater de $\log_G^\vee(\mu_b\mu_a^{-1})$ la moyenne de la distribution dans l'algèbre de Lie. Cependant, nous observons également un effet de distorsion géodésique, puisque la covariance a été transformée par la matrice $\varphi_G(\log_G^\vee(\mu_b\mu_a^{-1}))$. Nous parlons alors de *reparamétrisation* de la covariance. Une représentation graphique de ce phénomène est présenté en figure 1.6.

1.2.2.2 Interprétation géométrique

Dans le cas d'une distribution normale concentrée sur groupe de Lie $\mathcal{N}_G^R(X; \mu_b, P_b)$, l'information contenue dans la matrice de covariance P_b quantifie l'incertitude dans l'espace tangent à la moyenne μ_b . Donc l'information contenue dans P_b a une signification uniquement vis à vis de μ_b .

La matrice $\varphi_G(\log_G^\vee(\mu_b\mu_a^{-1}))$ permet de reparamétriser P_b afin de lui donner un *sens* non plus vis à vis de μ_b mais de μ_a .

La matrice $\Phi_G(\log_G^\vee(\mu_b\mu_a^{-1})) = \varphi_G(\log_G^\vee(\mu_b\mu_a^{-1}))^{-1}$ permet simplement de faire l'opération inverse.

Dans les chapitres suivants, lorsque nous verrons des matrices φ_G , ou bien leur inverse Φ_G , agir sur des matrices de covariances, il sera ainsi possible d'interpréter géométriquement l'effet de ces matrices.

1.2.3 Marginalisation

Nous nous intéressons maintenant à la marginalisation d'une distribution normale concentrée.

Considérons un groupe de Lie de la forme $G = G' \times G''$, où G' est un groupe de Lie matriciel de dimension p et G'' est un groupe de Lie matriciel de dimension q . Une distribution normale concentrée (à droite) peut alors s'écrire :

$$p(X) = p(X_1, X_2) \approx \frac{1}{\sqrt{(2\pi)^{p+q} \det(\Sigma)}} e^{-\frac{1}{2}Q(X_1, X_2)} \quad (1.115)$$

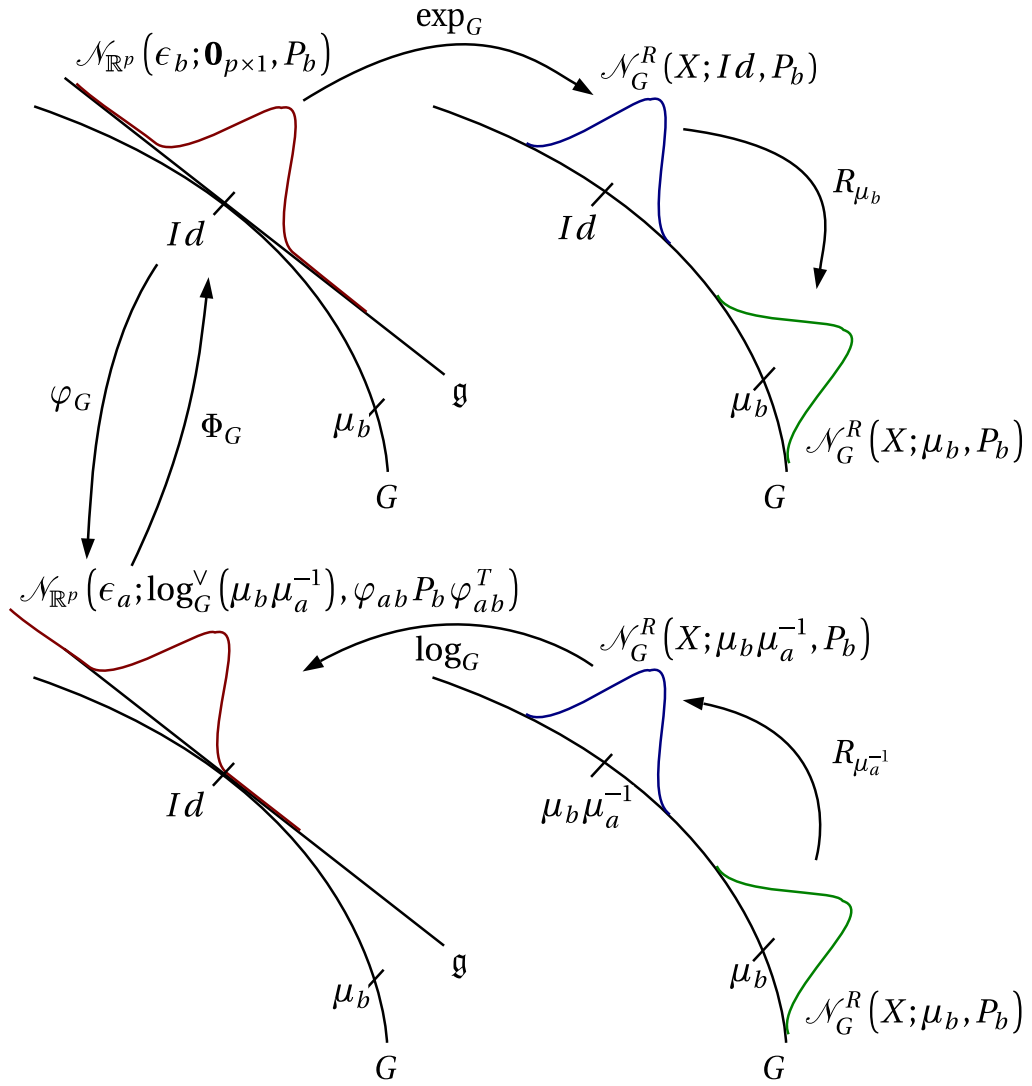


Figure 1.6 – Illustration graphique de la notion de reparamétrisation de la covariance

où $X \in G$, $X_1 \in G'$, $X_2 \in G''$ et

$$Q(X_1, X_2) = \left[\log_{G'}^\vee (X_1 \mu_1^{-1})^T, \log_{G''}^\vee (X_2 \mu_2^{-1})^T \right] \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}^{-1} \begin{bmatrix} \log_{G'}^\vee (X_1 \mu_1^{-1}) \\ \log_{G''}^\vee (X_2 \mu_2^{-1}) \end{bmatrix} \quad (1.116)$$

Ainsi, en tirant parti du fait que la distribution est concentrée, il est possible de montrer que (voir annexe A) :

$$p(X_1) \approx \mathcal{N}_{G'}^R(X_1; \mu_1, \Sigma_{11}) \quad (1.117)$$

1.2.4 Cas particulier de la distribution normale multivariée

La distribution *normale concentrée sur groupe de Lie* est une généralisation de la loi normale multivariée aux groupes de Lie matriciels. En effet, si l'on s'intéresse à la forme de (1.107) pour le cas particulier où G est un espace euclidien \mathbb{R}^p nous obtenons alors :

$$p(X) = \frac{1}{\sqrt{(2\pi)^p \det(P)}} e^{-\frac{1}{2} \|\log_G^\vee(\mu^{-1}X)\|_P^2} = \frac{1}{\sqrt{(2\pi)^p \det(P)}} e^{-\frac{1}{2} \|x-m\|_P^2} \quad (1.118)$$

où $X = \begin{bmatrix} Id_p & x \\ 0 & 1 \end{bmatrix}$ et $\mu = \begin{bmatrix} Id_p & m \\ 0 & 1 \end{bmatrix}$. Ici, la matrice Jacobienne à gauche vaut l'identité car le groupe est commutatif. Nous trouvons donc bien que $p(X)$ a la forme d'une la loi normale multivariée.

1.2.5 Cas particulier de la distribution log-normale

Lorsqu'on écrit la définition (1.97) de la distribution *normale concentrée sur groupe de Lie* pour le cas spécifique où G est le groupe multiplicatif (qui est commutatif), nous obtenons :

$$X = \mu \exp(\epsilon) = \exp(\log(\mu) + \epsilon) = \exp(\tau) \quad (1.119)$$

où $\tau = \log(\mu) + \epsilon$, \exp et \log sont les fonctions exponentielle et logarithme. Par conséquent $\tau \sim \mathcal{N}_{\mathbb{R}}(\tau; \log(\mu), \sigma^2)$ c'est à dire :

$$p(\tau) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\tau - \log(\mu))^2}{2\sigma^2}} \quad (1.120)$$

Or, par définition, si une variable τ suit une loi normale d'espérance $\log(\mu)$ et de variance σ^2 alors $X = \exp(\tau)$ suit une loi log-normale [Johnson 1994] :

$$p_{\ln}(X) = \frac{1}{X\sqrt{2\pi\sigma^2}} e^{-\frac{(\log(X)-\log(\mu))^2}{2\sigma^2}} \quad (1.121)$$

Si l'on s'intéresse maintenant à la forme de la distribution normale concentrée (1.107) pour le cas particulier de (\mathbb{R}^+*, \times) dont la matrice Jacobienne à gauche vaut l'identité (car le groupe est commutatif), nous obtenons alors :

$$p(X) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\log(\mu^{-1}X)^2}{2\sigma^2}} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\log(X)-\log(\mu))^2}{2\sigma^2}} \quad (1.122)$$

Alors que l'on pouvait légitimement s'attendre à ce que la distribution *normale concentrée sur groupe de Lie* soit une généralisation de la loi log-normale aux groupes de Lie matriciels, nous voyons ici que $p_{\ln}(X) \neq p(X)$. En réalité, $p_{\ln}(X)$ est une densité de probabilité par rapport à la mesure de Lebesgue $d_L X$:

$$\int_{(\mathbb{R}^+*, \times)} p_{\ln}(X) d_L X = 1 \quad (1.123)$$

alors que $p(X)$ est une densité de probabilité par rapport à la mesure de Haar $d_H X$:

$$\int_{(\mathbb{R}^+*, \times)} p(X) d_H X = 1 \quad (1.124)$$

Si l'on se souvient maintenant que (voir [Faraut 2008] paragraphe 5.2), pour le groupe multiplicatif :

$$d_H X = \frac{1}{X} d_L X \quad (1.125)$$

alors nous retrouvons bien :

$$\begin{aligned} \int_{(\mathbb{R}^+*, \times)} p(X) d_H X &= \int_{(\mathbb{R}^+*, \times)} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\log(X)-\log(\mu))^2}{2\sigma^2}} d_H X \\ &= \int_{(\mathbb{R}^+*, \times)} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\log(X)-\log(\mu))^2}{2\sigma^2}} \frac{1}{X} d_L X = \int_{(\mathbb{R}^+*, \times)} p_{\ln}(X) d_L X \end{aligned} \quad (1.126)$$

La distribution *normale concentrée sur groupe de Lie* est donc bien une généralisation de la loi log-normale aux groupes de Lie matriciels.

1.2.6 La distribution “en banane” sur $SE(2)$

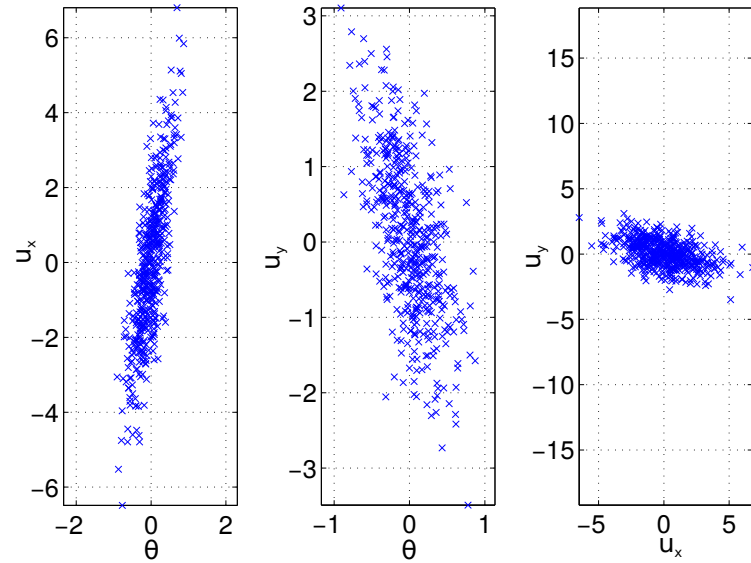
Nous nous intéressons ici à la distribution *normale concentrée* sur le groupe de Lie $SE(2)$ (voir paragraphe 1.1.6.5) dont la forme “en banane” [Long 2012] permet d'illustrer l'intérêt de cette distribution. En effet, l'illustration graphique de cette distribution proposée en figure 1.7 fait penser à l'incertitude que l'on

peut avoir lorsqu'on cherche à estimer la position et l'orientation d'un robot ou d'un véhicule en train d'effectuer un virage [Thrun 2005]. Remarquons que cette forme en banane provient de la non-linéarité de $\exp^{\wedge}_{SE(2)}$ qui fait agir la composante rotationnelle θ sur la composante positionnelle u (flèches bleues dans la figure 1.7). Cette propriété provient du fait que $SE(2)$ est le produit semi-direct de $SO(2)$ et \mathbb{R}^2 . Bien entendu, la forme en banane est accentuée par l'action à droite du groupe de Lie (flèches vertes dans la figure 1.7).

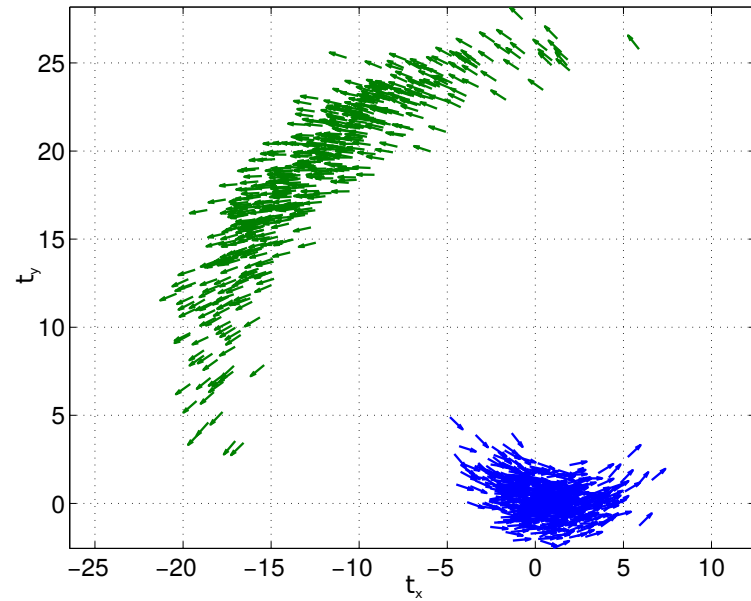
1.3 Résumé des contributions et conclusion

Ce chapitre nous a permis d'introduire les différents concepts liés au formalisme des groupes de Lie matriciels. Nous avons ensuite présenté une distribution définie pour un groupe de Lie matriciel, appelée distribution normale concentrée, qui sera utilisée tout au long du manuscrit dans un cadre d'estimation bayésienne.

La principale contribution du chapitre est le calcul de la dérivée seconde de l'application logarithme de matrice. Le chapitre en lui-même n'apporte pas d'autre contribution fondamentalement nouvelle. Il constitue cependant un recueil d'informations essentielles pour aborder le sujet de l'estimation de paramètres évoluant sur un groupe de Lie. Les concepts introduits proviennent de différents ouvrages qui sont cités tout au long du chapitre. Ils ont été réunis ici selon une approche intuitive, à travers des illustrations graphiques permettant d'interpréter géométriquement ces notions souvent abstraites.



(a) Échantillons dans l'algèbre de Lie $\mathfrak{se}(2)$ tirés selon la distribution : $\mathcal{N}_{\mathbb{R}^3}(\epsilon; \mathbf{0}_{3 \times 1}, P = \begin{bmatrix} 0.1 & 0.5 & -0.2 \\ 0.5 & 4 & -1 \\ -0.2 & -1 & 1 \end{bmatrix})$



(b) Échantillons transformés en utilisant $\exp_{SE(2)}^\wedge(\cdot)$ (flèches bleues) puis transportés autour de la “moyenne” $\mu = \exp_{SE(2)}^\wedge([\pi \ 30 \ 20]^T)$ en utilisant l'action à droite de $SE(2)$ sur lui-même (flèches vertes)

Figure 1.7 – Illustration graphique de la distribution *normale concentrée* sur le groupe de Lie $SE(2)$. La direction d'une flèche peut être interprétée comme l'orientation d'un véhicule alors que la base de la flèche correspond à sa position.

Filtrage de Kalman à temps discret et continu-discret sur groupes de Lie matriciels

2.1	Introduction	44
2.2	État de l'art	45
2.3	Filtre de Kalman étendu continu-discret	47
2.3.1	Rappels sur le filtrage de Kalman étendu continu-discret sur espace euclidien	47
2.3.2	Modèles de propagation et d'observation proposés	49
2.3.3	Solution proposée	50
2.3.4	Propagation	50
2.3.5	Mise à jour	55
2.3.6	Algorithme CD-LG-EKF	58
2.3.7	Application du CD-LG-EKF à un problème de filtrage de la trajectoire d'une caméra	59
2.4	Filtre de Kalman étendu discret	71
2.4.1	Rappels sur le filtrage de Kalman étendu discret sur espace euclidien	71
2.4.2	Modèles de propagation et d'observation proposés	73
2.4.3	Solution proposée	73
2.4.4	Propagation	73
2.4.5	Mise à jour	75
2.4.6	Algorithme D-LG-EKF	76
2.4.7	Application du D-LG-EKF à un problème de filtrage de la trajectoire d'une caméra	76
2.5	Résumé des contributions et conclusion	80

2.1 Introduction

Dans le contexte du filtrage bayésien, il existe deux grands types d’approches permettant d’estimer un état évoluant sur un espace euclidien à partir d’observations elles-mêmes euclidiennes : d’un côté les approches de type filtrage de Kalman, comme le filtre de Kalman étendu [Maybeck 1979], le filtre de Kalman sans parfum [Sarkka 2007] ou encore le filtre de Kalman Cubature [Arasaratna 2010], et d’un autre côté les approches de type filtrage particulière [Doucet 2001]. Seul un nombre restreint de travaux se sont intéressés à étendre ces approches au cadre des variétés (voir Tableau 2.1). Des filtres particuliers dédiés aux variétés riemanniennes [Snoussi 2006], aux variétés de Stiefel [Tompkins 2007] ou encore aux variétés de Grassmann [Rentmeeste 2010] ont été proposés, alors qu’un filtre de Kalman sans parfum permettant d’estimer un état évoluant sur une variété riemannienne à partir d’observations également riemanniennes a récemment été développé [Hauberg 2013].

Dans ce chapitre nous nous intéressons au développement de méthodes de filtrage de Kalman permettant d’estimer des paramètres évoluant sur des groupes de Lie.

Plusieurs approches sont envisageables. La plus simple est probablement de paramétrer le groupe de Lie. Cela revient à transformer les paramètres que l’on cherche à estimer, qui sont contraints à appartenir à un groupe de Lie, en des paramètres euclidiens. C’est par exemple ce que nous faisons lorsque nous définissons trois angles d’Euler pour représenter une rotation en dimension 3. Les paramètres obtenus étant euclidiens, il est alors possible d’utiliser un filtre de Kalman “classique”. Cependant, le fait de paramétrer un groupe de Lie produit des singularités. Dans le cas des angles d’Euler, cette singularité, désignée sous le nom de blocage de cardan, conduit à de mauvaises performances des filtres utilisés.

Une autre approche consiste à voir le problème comme un problème de filtrage euclidien avec des contraintes d’égalité forçant l’état à rester sur la variété. On parle alors d’approche extrinsèque. Dans ce cadre, il est possible d’utiliser des approches génériques de filtrage sous contrainte telles que celle proposée dans [Simon 2010]. Néanmoins, les matrices de covariances sont alors singulières ce qui peut poser des problèmes de stabilité numérique.

Ici, nous proposons d’employer une troisième approche qui consiste à prendre en compte la géométrie des groupes de Lie de manière intrinsèque. Comme nous le verrons par la suite, les algorithmes obtenus sont alors non seulement élégants mais également numériquement stables.

Dans le contexte du filtrage sur groupe de Lie, la plupart des modèles physiques sont décrits par des équations différentielles alors que les observations

sont discrètes. Par conséquent, dans une première partie, nous proposons une généralisation du filtre de Kalman étendu à temps continu-discret au cas où l'état et les observations évoluent sur des groupes de Lie. Plus précisément, la propagation de l'état à temps continu est régie par une équation différentielle stochastique sur groupe de Lie, alors que l'équation d'observation discrète relie non-linéairement l'état aux observations qui évoluent sur un autre groupe de Lie.

Dans une seconde partie, au lieu de chercher une solution à un problème continu-discret, nous nous intéressons au cas où l'équation différentielle stochastique est d'abord approchée par une équation de propagation discrète. Cela nous conduit alors à proposer une généralisation du filtre de Kalman étendu à temps discret au cas où l'état et les observations évoluent sur des groupes de Lie.

2.2 État de l'art

De nombreux travaux se sont intéressés à la prise en compte de la structure de groupe de Lie dans le contexte spécifique de l'estimation de l'orientation d'un aéronef. Parmi ces travaux, le filtre de Kalman étendu multiplicatif (MEKF) [Lefferts 1982; Markley 2003] correspond à une modification ad hoc du filtre de Kalman étendu continu-discret (CD-EKF) permettant de prendre en compte la contrainte de norme unitaire d'un quaternion unitaire. Plusieurs articles emploient ce formalisme [Mourikis 2009; Hall 2008; Trawny 2007; Trawny 2005].

Toujours dans le contexte de l'estimation d'une orientation 3D, des modifications ad hoc du filtre de Kalman sans parfum discret (UKF) ont également été proposées [Crassidis 2003; Kraft 2003; Sipos 2008].

Un filtre de Kalman étendu discret (EKF), appelé Motor EKF, dédié au groupe des moteurs représentant un double recouvrement du groupe de Lie $SE(3)$ [BayroCorro 2010], et utilisant le formalisme de l'algèbre géométrique, a été développé dans [BayroCorro 2000] afin d'estimer l'orientation et la position d'un solide.

On trouve également une présentation d'un algorithme, appelé "Invariant Momentum-tracking Kalman Filter", permettant d'estimer un quaternion unitaire ainsi qu'un moment angulaire dans [Persson 2012].

Le filtre de Kalman étendu invariant récemment proposé [Bonnabel 2009; Bonnabel 2007; Martin 2010; Barrau 2015; Hervier 2012] est proche des algorithmes que nous développons dans ce chapitre. Ce filtre à temps continu est dédié à des systèmes possédant des symétries. Il permet à la fois de prendre en compte intrinsèquement la géométrie du groupe de Lie sur lequel l'état

<i>Approche</i>	<i>Variété de l'état</i>	<i>Variété des observations</i>	<i>Type de système</i>	<i>Type de filtre</i>
[Bonnabel 2009]	Groupe de Lie	Espace euclidien	Continu possédant des symétries	EKF
[Snoussi 2006]	Variété Riemannienne	Espace euclidien	Discret	PF
[Tompkins 2007]	Variété de Stiefel	Espace euclidien	Discret	PF
[Hauberg 2013]	Variété Riemannienne	Variété Riemannienne	Discret	UKF
[Rentmeeste 2010]	Variété de Grassmann	Variété de Grassmann	Discret	PF
Paragraphe 2.4 de ce manuscrit	Groupe de Lie matriciel	Groupe de Lie matriciel	Discret	EKF
Paragraphe 2.3 de ce manuscrit	Groupe de Lie matriciel	Groupe de Lie matriciel	Continu-Discret	EKF

Tableau 2.1 – Récapitulatif des approches de type filtrage de Kalman et filtre particulière pour un état évoluant sur une variété. (PF = filtre particulière, EKF = filtre de Kalman étendu, UKF = filtre de Kalman sans parfum)

évolue et étend également le domaine de convergence du filtre de Kalman aux trajectoires dites “permanentes”. Toutefois, ni le problème de mesures discrètes, ni le problème d’observations évoluant sur un groupe de Lie n’est traité.

2.3 Filtre de Kalman étendu continu-discret

Contrairement aux approches de l’état de l’art, nous proposons dans ce paragraphe un algorithme capable d’estimer, de manière intrinsèque, un état vivant sur un groupe de Lie, dont l’évolution est décrite par une équation différentielle stochastique sur groupe de Lie, et ce à partir d’observations évoluant également sur un groupe de Lie.

En partant du principe que la distribution a posteriori de l’état est une distribution normale concentrée, nous obtenons des équations permettant de propager et mettre à jour efficacement la moyenne et la covariance de cette distribution. Nous montrons également que l’algorithme obtenu, appelé filtre de Kalman étendu continu-discret sur groupe de Lie (CD-LG-EKF), se réduit au traditionnel CD-EKF dans le cas où à la fois l’état et les observations évoluent sur des espaces euclidiens.

2.3.1 Rappels sur le filtrage de Kalman étendu continu-discret sur espace euclidien

Nous rappelons ici brièvement l’algorithme CD-EKF [Maybeck 1979].

2.3.1.1 Modèles de propagation et d’observation

L’algorithme CD-EKF permet d’estimer l’état $x(t) \in \mathbb{R}^p$ d’un système dynamique, décrit par une équation différentielle stochastique non linéaire, à partir d’observations discrètes reliées à l’état par une équation non linéaire :

$$\begin{cases} dx(t) &= \Omega(x(t)) dt + db(t) \\ z_k &= h(x_k) + w_k \end{cases} \quad (2.1)$$

où $x_k = x(t = t_k)$, $z_k = z(t = t_k) \in \mathbb{R}^q$ est une observation reçue à l’instant $t = t_k$ et $b(t)$ est un mouvement brownien de dimension p , de coordonnées $(b_1(t), \dots, b_p(t))$ et de matrice de covariance R . De plus, $w_k \sim \mathcal{N}_{\mathbb{R}^q}(w_k; \mathbf{0}, Q_k)$ est un bruit blanc gaussien alors que $\Omega(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}^p$ et $h(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}^q$ sont des fonctions différentiables potentiellement non linéaires.

2.3.1.2 Principe

Une manière d'interpréter le CD-EKF est de supposer qu'à chaque instant la distribution a posteriori de $x(t)$ est une distribution normale multivariée de moyenne $\mu(t)$ et de covariance $P(t)$. L'objectif du CD-EKF est alors d'estimer $\mu(t)$ et $P(t)$ à chaque instant sachant toutes les observations précédemment reçues. L'algorithme obtenu opère en deux étapes qui alternent. Supposons qu'à l'instant $t = t_{k-1}$ les paramètres $\mu_{k-1|k-1}$ et $P_{k-1|k-1}$ de la distribution $p(x_{k-1}|z_1, z_2, \dots, z_{k-1}) \approx \mathcal{N}_{\mathbb{R}^p}(x_{k-1}; \mu_{k-1|k-1}, P_{k-1|k-1})$ sont connus. Alors, jusqu'à la réception d'une nouvelle observation, c'est à dire de l'instant $t = t_{k-1}$ à l'instant $t = t_k$, la moyenne $\mu_{k-1|k-1}$ et la covariance $P_{k-1|k-1}$ sont propagées afin d'obtenir les paramètres $\mu_{k|k-1}$ et $P_{k|k-1}$ de la distribution $p(x_k|z_1, z_2, \dots, z_{k-1}) \approx \mathcal{N}_{\mathbb{R}^p}(x_k; \mu_{k|k-1}, P_{k|k-1})$. Ces deux paramètres sont alors mis à jour en utilisant l'information apportée par la nouvelle observation z_k pour obtenir les paramètres $\mu_{k|k}$ et $P_{k|k}$ de la distribution $p(x_k|z_1, z_2, \dots, z_{k-1}, z_k) \approx \mathcal{N}_{\mathbb{R}^p}(x_k; \mu_{k|k}, P_{k|k})$. Dans le cas du CD-EKF, les équations permettant de propager et de mettre à jour la moyenne et la covariance sont obtenues en linéarisant les modèles de propagation et d'observation.

2.3.1.3 Algorithme CD-EKF

L'algorithme du CD-EKF est présenté ci-après :

Algorithme 2.1 CD-EKF

Entrée : $\mu_{k-1|k-1}, P_{k-1|k-1}, z_k$

Sortie : $\mu_{k|k}, P_{k|k}$

Propagation sur $t \in [t_{k-1}, t_k]$:

Intégrer les équations différentielles

$$\frac{d\mu}{dt} = \Omega(\mu) \text{ où } \mu(t_{k-1}) = \mu_{k-1|k-1}$$

$$\frac{dP}{dt} = FP + PF^T + R \text{ où } P(t_{k-1}) = P_{k-1|k-1} \text{ et } F = \left. \frac{d}{dx}\Omega(x) \right|_{x=\mu}$$

Mise à jour à l'instant $t = t_k$:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + Q_k)^{-1} \text{ où } H_k = \left. \frac{d}{dx}h(x) \right|_{x=\mu_{k|k-1}}$$

$$\tilde{z}_k = z_k - h(\mu_{k|k-1})$$

$$m_{k|k}^- = K_k \tilde{z}_k$$

$$\mu_{k|k} = \mu_{k|k-1} + m_{k|k}^-$$

$$P_{k|k} = (Id - K_k H_k) P_{k|k-1}$$

Cet algorithme, qui est considéré comme une méthode standard dans le

domaine de l'estimation non linéaire, est restreint au cas où l'état et les observations évoluent sur des espaces euclidiens. Dans la suite de ce paragraphe, nous proposons d'étendre cette approche au cas où l'état et les observations évoluent sur des groupes de Lie.

2.3.2 Modèles de propagation et d'observation proposés

2.3.2.1 Modèle de propagation

Nous considérons le système dynamique suivant :

$$dX(t) = X(t) [\Omega(X(t))]_G^\wedge dt + \sum_{i=1}^p X(t) E_i \circ dB_i(t) \quad (2.2)$$

où $X(t) \in G$ est l'état que nous souhaitons estimer à l'instant t , G est un groupe de Lie matriciel de dimension p , les E_i sont les matrices formant une base de l'algèbre de Lie et $B(t)$ est un mouvement brownien de dimension p , de coordonnées $(B_1(t), \dots, B_p(t))$, et de matrice de covariance R , c'est à dire $B(t_1) - B(t_2) \sim \mathcal{N}_{\mathbb{R}^p}(B(t_1) - B(t_2); \mathbf{0}_{p \times 1}, (t_2 - t_1)R)$ avec $t_1 < t_2$ (voir [Said 2012]). La notation \circ indique que nous utilisons l'intégrale de Stratanovich. La fonction $\Omega(X) : G \mapsto \mathbb{R}^p$ est différentiable et potentiellement non linéaire.

D'un point de vue plus intuitif (mais mathématiquement moins juste), l'équation (2.2) peut être réécrite de la façon suivante :

$$\dot{X}(t) = X(t) \left([\Omega(X(t))]_G^\wedge + [n(t)]_G^\wedge \right) \quad (2.3)$$

où $n(t) \sim \mathcal{N}_{\mathbb{R}^p}(n(t); \mathbf{0}_{p \times 1}, R)$ est un bruit blanc gaussien et $[n(t)]_G^\wedge \in \mathfrak{g}$. Ainsi $\dot{X}(t) \in T_X G$. Le bruit blanc injecté correspond donc ici à :

$$\sum_{i=1}^p X(t) E_i \circ dB_i(t) = [n(t)]_G^\wedge dt \quad (2.4)$$

Nous n'avons à aucun moment supposé que le système considéré est invariant par rapport à l'action d'un groupe de Lie contrairement à [Bonnabel 2009]. De plus, le fait que $\Omega(\cdot)$ soit potentiellement non linéaire va nous permettre par la suite de gérer des modèles de propagation complexes. Par exemple, dans nos simulations, nous considérons un groupe de Lie composé à la fois de la pose de la caméra et de sa vitesse (incluant la vitesse de rotation et la vitesse de translation), cette dernière étant utilisée lors de l'étape de propagation pour guider la pose.

Par la suite, sauf en cas d'ambiguïté, et afin de faciliter la lecture, le paramètre de temps t sera omis. Par exemple $X(t)$ sera noté simplement X .

Remarquons que si le groupe de Lie G sur lequel évolue l'état X est un espace euclidien, alors (2.2) se réduit au modèle de propagation du CD-EKF (voir équation (2.1)).

2.3.2.2 Modèle d'observation

Nous considérons également des observations discrètes vivant sur un groupe de Lie matriciel G' de dimension q :

$$z_k = h(X_k) \exp_{G'}^\wedge(w_k) \quad (2.5)$$

où $h : G \rightarrow G'$ est une fonction différentiable, $z_k = z(t_k) \in G'$, $X_k = X(t_k) \in G$ est l'état que nous souhaitons estimer au temps t_k et $w_k \sim \mathcal{N}_{\mathbb{R}^q}(w_k; \mathbf{0}_{q \times 1}, Q_k)$ est un bruit blanc gaussien.

Remarquons que dans le cas où G et G' sont des espaces euclidiens, alors (2.5) se réduit au modèle d'observation du CD-EKF (voir équation (2.1)).

2.3.3 Solution proposée

Afin d'estimer l'état X , nous proposons de supposer que sa distribution a posteriori a la forme d'une distribution normale concentrée (à gauche) :

$$p(X_k | z_1, \dots, z_l) \approx \mathcal{N}_G^L(X_k; \mu_{k|l}, P_{k|l}) \quad (2.6)$$

Plus précisément, nous nous intéressons aux cas où $l = k - 1$ (étape de propagation) et $l = k$ (étape de mise à jour). Ainsi, notre objectif est de montrer comment propager et mettre à jour les paramètres μ et P de la distribution normale concentrée. Dans notre formalisme, nous choisissons comme estimateur de l'état X la “moyenne” μ de la distribution normale concentrée a posteriori.

2.3.4 Propagation

Commençons par supposer que :

$$p(X_{k-1} | z_1, \dots, z_{k-1}) \approx \mathcal{N}_G^L(X_{k-1}; \mu_{k-1|k-1}, P_{k-1|k-1}) \quad (2.7)$$

où $\mu_{k-1|k-1}$ et $P_{k-1|k-1}$ sont connus, et

$$p(X_k | z_1, \dots, z_{k-1}) \approx \mathcal{N}_G^L(X_k; \mu_{k|k-1}, P_{k|k-1}) \quad (2.8)$$

où $\mu_{k|k-1}$ et $P_{k|k-1}$ sont inconnus. L'objectif de l'étape de propagation est de montrer comment propager $\mu_{k-1|k-1}$ et $P_{k-1|k-1}$ en utilisant le modèle de propagation (2.2) afin d'obtenir $\mu_{k|k-1}$ et $P_{k|k-1}$.

La définition de l'erreur dans l'algèbre de Lie $\epsilon(t)$ sera importante pour le reste des calculs :

$$\epsilon = [\epsilon_\wedge]_G^\vee = [\log_G(\eta)]_G^\vee \quad (2.9)$$

où η représente l'erreur dans le groupe :

$$\eta = \mu^{-1}X \quad (2.10)$$

Nous rappelons également que puisque (2.7) a la forme d'une distribution normale concentrée à gauche, X_{k-1} peut alors s'exprimer de la manière suivante :

$$X_{k-1} = \mu_{k-1|k-1} \exp_G^\wedge(\epsilon_{k-1}) \quad (2.11)$$

où l'erreur dans l'algèbre de Lie ϵ_{k-1} est distribuée de la manière suivante :

$$p(\epsilon_{k-1}|z_1, \dots, z_{k-1}) = \mathcal{N}_{\mathbb{R}^p}(\epsilon_{k-1}; m_{k-1|k-1} = 0, P_{k-1|k-1}) \quad (2.12)$$

Remarquons qu'une façon de propager $\mu_{k-1|k-1}$ et $P_{k-1|k-1}$ serait de tirer des trajectoires selon (2.2) puis de calculer leur barycentre et la covariance comme cela a été proposé dans [Park 2008]. Cependant, ici, nous souhaitons obtenir des formules explicites (approchées) permettant de propager la moyenne et la covariance et ce afin de proposer une généralisation du CD-EKF aux groupes de Lie. De plus, le fait de tirer un nombre suffisant de trajectoires permettrait bien d'obtenir des résultats plus précis que ceux produits par notre approche. Néanmoins le coût calculatoire serait beaucoup plus élevé par la combinaison du coût élevé de l'application exponentielle avec un nombre important de tirages.

Dans le cas spécifique où la fonction Ω ne dépend pas de $X(t)$, c'est à dire qu'elle soit constante ou qu'elle soit une fonction explicite de t , alors [Wang 2008] propose de générer des trajectoires sur de petits intervalles. Ensuite, la moyenne et la covariance sont calculées pour chaque intervalle. La moyenne et la covariance de la trajectoire complète sont finalement obtenues en concaténant les moyennes et covariances estimées pour chaque petit intervalle en utilisant les équations proposées dans leur article. Dans notre cas, nous souhaitons pouvoir gérer des modèles de propagation complexes et nous traitons donc le cas où Ω dépend de $X(t)$.

2.3.4.1 Propagation de la moyenne

Nous choisissons de propager la moyenne $\mu(t)$ en utilisant le modèle d'état (2.2) sans bruit :

$$d\mu = \mu [\Omega(\mu)]_G^\wedge dt \quad (2.13)$$

où $\frac{d\mu}{dt} \in T_\mu G$.

Nous montrons par la suite qu'en propageant la moyenne de cette manière, la moyenne de l'erreur dans l'algèbre de Lie $m(t) = \mathbb{E}[\epsilon(t)]$ demeure nulle au premier ordre.

Remarquons que si G est un espace euclidien, alors l'équation de propagation de la moyenne devient :

$$\frac{d\mu}{dt} = \Omega(\mu) \quad (2.14)$$

ce qui correspond à l'équation de propagation de la moyenne du CD-EKF (voir Alg. 2.1).

2.3.4.2 Propagation de l'erreur dans l'algèbre de Lie

Nous souhaitons maintenant expliciter le terme $d\epsilon(t)$ afin, à la fois de justifier (2.13) et également d'obtenir une équation de propagation pour la covariance $P(t)$. En utilisant (2.13) et le fait que $d(\mu^{-1}\mu) = d(\mu^{-1})\mu + \mu^{-1}d\mu$ nous avons :

$$d(\mu^{-1}) = -\mu^{-1}d\mu\mu^{-1} = -[\Omega(\mu)]_G^\wedge dt\mu^{-1} \quad (2.15)$$

Donc d'après (2.15) et (2.2) :

$$\begin{aligned} d\eta &= d(\mu^{-1}X) = d(\mu^{-1})X + \mu^{-1}dX \\ &= -[\Omega(\mu)]_G^\wedge dt\mu^{-1}X + \mu^{-1}\left(X[\Omega(X)]_G^\wedge dt + \sum_{i=1}^p XE_i \circ dB_i\right) \\ &= -[\Omega(\mu)]_G^\wedge dt\eta + \eta[\Omega(X)]_G^\wedge dt + \eta\sum_{i=1}^p E_i \circ dB_i \\ &= \eta\left(-\eta^{-1}[\Omega(\mu)]_G^\wedge \eta + [\Omega(X)]_G^\wedge\right) dt + \eta\sum_{i=1}^p E_i \circ dB_i \\ &= \eta\left(-[\text{Ad}_G(\eta^{-1})\Omega(\mu)]_G^\wedge dt + [\Omega(X)]_G^\wedge dt + dB\right) \end{aligned} \quad (2.16)$$

où

$$dB = \sum_{i=1}^p E_i \circ dB_i \quad (2.17)$$

Prenons maintenant le développement de Taylor à l'ordre 1 de Ω :

$$\begin{aligned} \Omega(X) &= \Omega(\mu \exp_G([\epsilon]_G^\wedge)) \\ &= \Omega(\mu) + F\epsilon + O(\|\epsilon\|^2) \end{aligned} \quad (2.18)$$

où

$$F = \frac{d}{d\delta} \Omega(\mu \exp_G([\delta]_G^\wedge)) \Big|_{\delta=0} \quad (2.19)$$

A partir des deux résultats précédents, du fait que $\eta = \exp_G^\wedge(\epsilon)$ et que $\text{Ad}_G(\exp_G^\wedge(\cdot)) = \text{expm}(\text{ad}_G(\cdot))$, nous déduisons l'expression de $d\xi(t)$:

$$\begin{aligned}
 d\xi &= \eta^{-1} \circ d\eta \\
 &= - \left[\text{Ad}_G(\eta^{-1}) \Omega(\mu) \right]_G^\wedge dt + [\Omega(X)]_G^\wedge dt + dB \\
 &= - \left[\text{Ad}_G(\exp_G^\wedge(-\epsilon)) \Omega(\mu) \right]_G^\wedge dt + [\Omega(X)]_G^\wedge dt + dB \\
 &= - \left[\text{expm}(-\text{ad}_G(\epsilon)) \Omega(\mu) \right]_G^\wedge dt + [\Omega(X)]_G^\wedge dt + dB \\
 &= - \left[\left(Id - \text{ad}_G(\epsilon) + O(\|\epsilon\|^2) \right) \Omega(\mu) \right]_G^\wedge dt + \left[\Omega(\mu) + F\epsilon + O(\|\epsilon\|^2) \right]_G^\wedge dt + dB \\
 &= \left[\left(\text{ad}_G(\epsilon) \Omega(\mu) + F\epsilon + O(\|\epsilon\|^2) \right) \right]_G^\wedge dt + dB \\
 &= [(F - \text{ad}_G(\Omega(\mu))) \epsilon]_G^\wedge dt + O(\|\epsilon\|^2) dt + dB \tag{2.20} \\
 &= \mathcal{L}\epsilon_\wedge dt + O(\|\epsilon_\wedge\|^2) dt + dB \tag{2.21}
 \end{aligned}$$

où $[\epsilon]_G^\wedge = \epsilon_\wedge$ et

$$\mathcal{L}\epsilon_\wedge = [(F - \text{ad}_G(\Omega(\mu))) \epsilon]_G^\wedge \tag{2.22}$$

Afin d'obtenir l'expression de $d\epsilon_\wedge(t) = d\log_G(\eta)$, nous utilisons le lemme d'Itô sur groupe de Lie (voir [Arnaudon 1992]) qui, dans notre cas, requiert les expressions des dérivées première et seconde du logarithme de matrice (voir paragraphe 1.1.5). Nous obtenons ainsi :

$$\begin{aligned}
 d\epsilon_\wedge &= d\log(\eta) = \frac{d}{ds} \log_G(\eta \exp_G(sd\xi)) \Big|_{s=0} + \frac{1}{2} \frac{d^2}{ds^2} \log_G(\eta \exp_G(sd\xi)) \Big|_{s=0} \\
 &= d\xi + \frac{1}{2} [\epsilon_\wedge, d\xi] + \frac{1}{12} [\epsilon_\wedge, [\epsilon_\wedge, d\xi]] \\
 &\quad + \frac{1}{12} [d\xi, [d\xi, \epsilon_\wedge]] + O(\|\epsilon_\wedge\|^2) (d\xi, d\xi) + O(\|\epsilon_\wedge\|^3) d\xi \\
 &= \mathcal{L}\epsilon_\wedge dt + dB + \frac{1}{2} [\epsilon_\wedge, dB] + \frac{1}{12} [\epsilon_\wedge, [\epsilon_\wedge, dB]] \\
 &\quad + \frac{1}{12} [dB, [dB, \epsilon_\wedge]] + O(\|\epsilon_\wedge\|^2) dt + O(\|\epsilon_\wedge\|^3) dB \tag{2.23}
 \end{aligned}$$

Ce qui nous donne finalement :

$$\begin{aligned}
 d\epsilon &= (F - \text{ad}_G(\Omega(\mu))) \epsilon dt + db + \frac{1}{2} \text{ad}_G(\epsilon) db \\
 &\quad + \frac{1}{12} \text{ad}_G(\epsilon)^2 db + \frac{1}{12} \text{ad}_G(db)^2 \epsilon + O(\|\epsilon_\wedge\|^2) dt + O(\|\epsilon_\wedge\|^3) db \\
 &= \left(F - \text{ad}_G(\Omega(\mu)) + \frac{1}{12} C(R) \right) \epsilon dt \\
 &\quad + \left(Id + \frac{1}{2} \text{ad}_G(\epsilon) + \frac{1}{12} \text{ad}_G(\epsilon)^2 \right) db + O(\|\epsilon\|^2) dt + O(\|\epsilon\|^3) db \tag{2.24}
 \end{aligned}$$

où

$$db = \sum_{i=1}^p e_i \circ dB_i \quad (2.25)$$

et où (voir Annexe C)

$$(C(R)dt)_{ij} = \left(\text{ad}_G(db)^2 \right)_{ij} = \sum_{k=1}^p L_{ik}^T db db^T L_{kj} \quad (2.26)$$

avec

$$db db^T = \left(\sum_{l=1}^p e_l dB_l \right) \left(\sum_{m=1}^p e_m^T dB_m \right) \quad (2.27)$$

$$= \sum_{l=1}^p \sum_{m=1}^p e_l e_m^T dB_l dB_m = \sum_{l=1}^p \sum_{m=1}^p e_l e_m^T R_{lm} dt \quad (2.28)$$

En négligeant les termes en $O(\|\epsilon\|^2)dt$ et en $O(\|\epsilon\|^3)db$ dans (2.24), la moyenne $m(t)$ de $\epsilon(t)$ reste nulle au cours de la propagation si $m(t_0) = \mathbf{0}$ (voir paragraphe D) :

$$\frac{dm}{dt} = Jm \quad (2.29)$$

où

$$J = F - \text{ad}_G(\Omega(\mu)) + \frac{1}{12}C(R) \quad (2.30)$$

2.3.4.3 Propagation de la covariance

A nouveau en négligeant les termes en $O(\|\epsilon\|^2)dt$ et en $O(\|\epsilon\|^3)db$, l'équation de propagation de la covariance associée à l'équation (2.24) est alors (voir paragraphe D) :

$$\begin{aligned} \frac{dP}{dt} &= JP + PJ^T + R + \frac{1}{4}\mathbb{E}\left(\text{ad}_G(\epsilon) R \text{ad}_G(\epsilon)^T\right) \\ &\quad + \frac{1}{12}\mathbb{E}\left(\text{ad}_G(\epsilon)^2\right) R + \frac{1}{12}R\mathbb{E}\left(\text{ad}_G(\epsilon)^2\right)^T \end{aligned} \quad (2.31)$$

où (voir paragraphe C) :

$$\mathbb{E}\left(\text{ad}_G(\epsilon)^2\right)_{ij} = \sum_{k=1}^p L_{ik}^T P L_{kj} \quad (2.32)$$

et

$$\mathbb{E}\left(\text{ad}_G(\epsilon) R \text{ad}_G(\epsilon)^T\right)_{ij} = \sum_{k=1}^p \sum_{l=1}^p R_{kl} L_{ik}^T P L_{jl} \quad (2.33)$$

Remarquons que si G est un espace euclidien, alors $\text{ad}_G = \mathbf{0}_{p \times p}$ et nous retrouvons bien l'équation de propagation de la covariance du CD-EKF (voir Alg.

2.1).

2.3.4.4 Résumé de l'étape de propagation

L'étape de propagation consiste à intégrer les équations différentielles (2.13) et (2.31), en partant des conditions initiales $\mu(t = t_{k-1}) = \mu_{k-1|k-1}$ et $P(t = t_{k-1}) = P_{k-1|k-1}$ jusqu'au temps t_k afin d'obtenir la moyenne $\mu_{k|k-1}$ et la covariance $P_{k|k-1}$. A la fin de l'étape de propagation la distribution a posteriori de l'état est alors paramétrée de la manière suivante :

$$p(X_k | z_1, \dots, z_{k-1}) \approx \mathcal{N}_G^L(X_k; \mu_{k|k-1}, P_{k|k-1}) \quad (2.34)$$

2.3.5 Mise à jour

Supposons maintenant que :

$$p(X_k | z_1, \dots, z_k) \approx \mathcal{N}_G^L(X_k; \mu_{k|k}, P_{k|k}) \quad (2.35)$$

Or à l'issue de l'étape de propagation nous avons :

$$p(X_k | z_1, \dots, z_{k-1}) \approx \mathcal{N}_G^L(X_k; \mu_{k|k-1}, P_{k|k-1}) \quad (2.36)$$

Par conséquent, l'objectif de cette étape de mise à jour est de montrer comment incorporer l'information contenue dans z_k en utilisant le modèle d'observation (2.5) pour mettre à jour $\mu_{k|k-1}$ et $P_{k|k-1}$ et ainsi obtenir $\mu_{k|k}$ et $P_{k|k}$.

2.3.5.1 Mise à jour de l'erreur dans l'algèbre de Lie

Tout d'abord, à partir du modèle d'observation (2.5), nous avons :

$$\mathbf{0} = \log_{G'}^\vee \left(\exp_{G'}^\wedge(-w_k) h(X_k)^{-1} z_k \right) \quad (2.37)$$

En linéarisant l'équation précédente autour de l'estimée courante de l'état, nous obtenons alors :

$$\mathbf{0} = -w_k + \log_{G'}^\vee \left(h(\mu_{k|k-1})^{-1} z_k \right) - H_k \epsilon_k + O(\|\epsilon_k\|^2, \|w_k\|^2) \quad (2.38)$$

où

$$H_k = - \left. \frac{\partial}{\partial \delta} \log_{G'}^\vee \left(h(\mu_{k|k-1} \exp_G^\wedge(\delta))^{-1} z_k \right) \right|_{\delta=\mathbf{0}} \quad (2.39)$$

Ainsi, en définissant le terme d'innovation $\tilde{z}_k = \log_{G'}^\vee \left(h \left(\mu_{k|k-1} \right)^{-1} z_k \right)$, nous avons :

$$\tilde{z}_k = H_k \epsilon_k + w_k + O \left(\|\epsilon_k\|^2, \|w_k\|^2 \right) \quad (2.40)$$

En négligeant les termes en $O \left(\|\epsilon_k\|^2, \|w_k\|^2 \right)$, l'équation (2.40) est linéaire en ϵ_k qui évolue sur \mathbb{R}^p . De plus, puisqu'à l'issue de l'étape de propagation nous avons (2.36), alors par définition de la distribution normale concentrée :

$$p \left(\epsilon_k | z_1, \dots, z_{k-1} \right) \approx \mathcal{N}_{\mathbb{R}^p} \left(\epsilon_k; m_{k|k-1} = \mathbf{0}_{p \times 1}, P_{k|k-1} \right) \quad (2.41)$$

Nous pouvons donc appliquer l'étape de mise à jour du CD-EKF (voir Alg. 2.1) pour mettre à jour $m_{k|k-1}$ et $P_{k|k-1}$ et ainsi obtenir les paramètres mis à jours $m_{k|k}^-$ et $P_{k|k}^-$:

$$\begin{cases} K_k = P_{k|k-1} H_k^T \left(H_k P_{k|k-1} H_k^T + Q_k \right)^{-1} \\ \tilde{z}_k = \log_{G'}^\vee \left(h \left(\mu_{k|k-1} \right)^{-1} z_k \right) \\ m_{k|k}^- = \mathbf{0}_{p \times 1} + K_k \tilde{z}_k \\ P_{k|k}^- = (Id - K_k H_k) P_{k|k-1} \end{cases} \quad (2.42)$$

Remarquons que si G et G' sont des espaces euclidiens, alors le terme d'innovation \tilde{z}_k correspond à celui du CD-EKF (voir Alg. 2.1).

2.3.5.2 Reparamétrisation de l'état

A la fin de l'étape de mise à jour, nous nous attendons à avoir une distribution normale concentrée :

$$X_k = \mu_{k|k} \exp_G^\wedge (\epsilon_k) \quad (2.43)$$

avec

$$p \left(\epsilon_k | z_1, \dots, z_k \right) = \mathcal{N}_{\mathbb{R}^p} \left(\epsilon_k; m_{k|k} = \mathbf{0}_{p \times 1}, P_{k|k} \right) \quad (2.44)$$

Au lieu de ça, nous avons une distribution de la forme :

$$X_k = \mu_{k|k-1} \exp_G^\wedge (\epsilon_k^-) \quad (2.45)$$

avec

$$p \left(\epsilon_k^- | z_1, \dots, z_k \right) = \mathcal{N}_{\mathbb{R}^p} \left(\epsilon_k^-; m_{k|k}^- \neq \mathbf{0}_{p \times 1}, P_{k|k}^- \right) \quad (2.46)$$

Par conséquent, nous proposons d'effectuer la reparamétrisation suivante :

$$\begin{aligned} X_{k|k} &= \mu_{k|k-1} \exp_G^\wedge(\epsilon_k^-) \\ &= \mu_{k|k-1} \exp_G^\wedge(m_{k|k}^- + r_{k|k}^-) \end{aligned} \quad (2.47)$$

où $r_{k|k}^- \sim \mathcal{N}_{\mathbb{R}^p}(r_{k|k}^-; \mathbf{0}_{p \times 1}, P_{k|k}^-)$. En utilisant (1.29), nous obtenons :

$$\begin{aligned} X_k &= \mu_{k|k-1} \exp_G^\wedge(m_{k|k}^-) \exp_G^\wedge\left(\Phi_G(-m_{k|k}^-) r_{k|k}^- + O\left(\|r_{k|k}^-\|^2\right)\right) \\ &= \mu_{k|k} \exp_G^\wedge(\epsilon_k) \end{aligned} \quad (2.48)$$

où

$$\mu_{k|k} = \mu_{k|k-1} \exp_G^\wedge(m_{k|k}^-) \quad (2.49)$$

et

$$\epsilon_k = \Phi_G(-m_{k|k}^-) r_{k|k}^- + O\left(\|r_{k|k}^-\|^2\right) \quad (2.50)$$

Ainsi en négligeant les termes en $O\left(\|r_{k|k}^-\|^2\right)$, nous avons alors :

$$m_{k|k} = \mathbb{E}[\epsilon_k] = \mathbf{0}_{p \times 1} \quad (2.51)$$

$$\begin{aligned} P_{k|k} &= \mathbb{E}[\epsilon_k \epsilon_k^T] \\ &= \mathbb{E}\left[\Phi_G(-m_{k|k}^-) r_{k|k}^- (r_{k|k}^-)^T \Phi_G(-m_{k|k}^-)^T\right] \\ &= \Phi_G(-m_{k|k}^-) P_{k|k}^- \Phi_G(-m_{k|k}^-)^T \end{aligned} \quad (2.52)$$

Nous voyons ici apparaître une reparamétrisation de la covariance pour laquelle nous avons proposé une interprétation géométrique dans le paragraphe 1.2.2. En effet, la matrice $P_{k|k}^-$ qui a un *sens* vis à vis de $\mu_{k|k-1}$ se voit reparamétrée par la matrice $\Phi_G(-\log_G^\vee(\mu_{k|k} \mu_{k|k-1}^{-1}))$ afin qu'elle ait un *sens* par rapport à la moyenne mise à jour $\mu_{k|k}$.

Remarquons que si G est un espace euclidien, alors $\Phi_G(\cdot) = Id$ et donc (2.42), (2.49) et (2.52) correspondent bien à la mise à jour du CD-EKF (voir Alg. 2.1).

2.3.5.3 Résumé de l'étape de mise à jour

L'étape de mise à jour nous a permis d'incorporer l'information contenue dans l'observation z_k afin de mettre à jour les paramètres $\mu_{k|k-1}$ et $P_{k|k-1}$ pour finalement obtenir $\mu_{k|k}$ et $P_{k|k}$. A la fin de l'étape de mise à jour la distribution

a posteriori de l'état est alors paramétrée de la manière suivante :

$$p(X_k | z_1, \dots, z_k) \approx \mathcal{N}_G^L(X_k; \mu_{k|k}, P_{k|k}) \quad (2.53)$$

2.3.6 Algorithme CD-LG-EKF

L'algorithme formé par les étapes de propagation et de mise à jour présentées précédemment est appelé filtre de Kalman étendu continu-discret sur groupe de Lie (CD-LG-EKF).

2.3.6.1 Vue générale

L'algorithme CD-LG-EKF est résumé ci-après (voir paragraphe 2.3.4 et paragraphe 2.3.5 pour plus de détails) :

Algorithme 2.2 CD-LG-EKF

Entrée : $\mu_{k-1|k-1}, P_{k-1|k-1}, z_k$

Sortie : $\mu_{k|k}, P_{k|k}$

Propagation sur $t \in [t_{k-1}, t_k]$:

Intégrer les équations différentielles suivantes :

$$\begin{aligned} \frac{d\mu(t)}{dt} &= \mu(t) [\Omega(\mu(t))]_G^\wedge \text{ où } \mu(t_{k-1}) = \mu_{k-1|k-1} \\ \frac{dP(t)}{dt} &= J(t)P(t) + P(t)J(t)^T + R + \frac{1}{4}\mathbb{E}\left(\text{ad}_G(\epsilon(t))\text{Rad}_G(\epsilon(t))^T\right) \\ &+ \frac{1}{12}\mathbb{E}\left(\text{ad}_G(\epsilon(t))^2\right)R + \frac{1}{12}R\mathbb{E}\left(\text{ad}_G(\epsilon(t))^2\right)^T \text{ où } P(t_{k-1}) = P_{k-1|k-1} \end{aligned}$$

Mise à jour à l'instant $t = t_k$:

$$\begin{aligned} K_k &= P_{k|k-1}H_k^T \left(H_k P_{k|k-1} H_k^T + Q_k \right)^{-1} \\ \tilde{z}_k &= \log_G^\vee \left(h \left(\mu_{k|k-1} \right)^{-1} z_k \right) \\ m_{k|k}^- &= K_k \tilde{z}_k \\ \mu_{k|k} &= \mu_{k|k-1} \exp_G^\wedge \left(m_{k|k}^- \right) \\ P_{k|k} &= \Phi_G \left(-m_{k|k}^- \right) (Id - K_k H_k) P_{k|k-1} \Phi_G \left(-m_{k|k}^- \right)^T \end{aligned}$$

2.3.6.2 Détails d'implémentation

Afin d'implémenter l'algorithme CD-LG-EKF, les équations (2.13) et (2.31) doivent être intégrées pendant une durée Δt où Δt représente le temps écoulé entre deux observations. Or $\mu(t)$ et $P(t)$ vivent respectivement sur un groupe de Lie G et la variété riemannienne des matrices symétriques définies positives Sym^+ . Nous proposons ici une technique d'intégration numérique permettant d'assurer qu'à la fois $\mu(t)$ et $P(t)$ se propagent sur leurs variétés respectives.

Propagation de la moyenne Pour propager $\mu(t) = \mu_t$, nous utilisons une méthode de Lie-Euler en faisant de “petits” pas δt , i.e $\delta t \ll \Delta t$:

$$\mu_{t+\delta t} \simeq \mu_t \exp_G^\wedge(\Omega(\mu_t) \delta t) \quad (2.54)$$

Propagation de la covariance Pour propager $P(t)$, nous avons choisi la métrique Log-Euclidienne [Arsigny 2006] où nous effectuons l’approximation suivante (où $f : \text{Sym}^+ \rightarrow \text{Sym}$) :

$$\frac{d \log m(P)}{dP}(f(P)) \simeq \frac{\log m(P + hf(P)) - \log m(P)}{h} \quad (2.55)$$

La covariance peut alors être propagée de la façon suivante en faisant de “petits” pas δt , i.e $\delta t \ll \Delta t$:

$$\begin{aligned} P_{t+\delta t} &\simeq \exp m \left(\log m(P_t) + \frac{d \log m(P_t)}{dP_t}(f(P_t)) \delta t \right) \\ &\simeq \exp m \left(\frac{1}{\alpha} (\log m(P_t + \alpha f(P_t) \delta t) - (1 - \alpha) \log m(P_t)) \right) \end{aligned} \quad (2.56)$$

De plus, en notant respectivement λ_{\min} et λ_{\max} les valeurs propres minimum et maximum de P_t , si $\lambda_{\min}(P_t) > \lambda_{\max}(\alpha f(P_t) \delta t)$ alors $(P_t + \alpha f(P_t) \delta t) \in \text{Sym}^+$. Par conséquent, nous avons la contrainte suivante sur α pour que $P_{t+\delta t}$ reste sur Sym^+ :

$$\alpha < \frac{\lambda_{\min}(P_t)}{\lambda_{\max}(f(P_t)) \delta t} \quad (2.57)$$

2.3.7 Application du CD-LG-EKF à un problème de filtrage de la trajectoire d’une caméra

Nous avons choisi d’appliquer le CD-LG-EKF au problème de l’estimation de la pose d’une caméra. Plus précisément, nous considérons un modèle de propagation où l’accélération est un bruit blanc et un modèle de mesure où une observation correspond à la pose de la caméra bruitée. Afin de réduire la taille des équations, nous utilisons dans ce sous-chapitre une notation symbolique au lieu d’une représentation matricielle utilisée jusqu’à présent. Ainsi, un élément du groupe de Lie sera noté $(\cdot)_G$ alors qu’un élément de son algèbre de Lie sera noté $(\cdot)_g$. Ces notations seront précisées à la page suivante.

2.3.7.1 Obtention des équations du filtre

Modèle de mouvement Nous considérons le modèle de mouvement suivant (nous utilisons les notations “intuitives” comme dans (2.3) permettant des

expressions plus simples. L'ensemble du raisonnement peut être transposé aux notations plus précises similaires à (2.2) :

$$\begin{cases} \dot{R} = R \left([\omega]_{SO(3)}^\wedge + [n_R]_{SO(3)}^\wedge \right) \\ \dot{\omega} = n_\omega \\ \dot{T} = v + n_T \\ \dot{v} = n_v \end{cases} \quad (2.58)$$

où

- $R = R_{gb} \in SO(3)$, représente l'orientation du référentiel global (g) par rapport au référentiel de la caméra (b)
- $\omega = \omega_b \in \mathbb{R}^3$ est la vitesse angulaire dans le référentiel de la caméra
- $T = \overrightarrow{O_g O_b}^g \in \mathbb{R}^3$ est la position de la caméra dans le référentiel global
- $v = v_g \in \mathbb{R}^3$ est la vitesse radiale dans le référentiel global
- n_R, n_ω, n_T et n_v sont des bruits blancs gaussiens (potentiellement corrélés)

Modèle d'observation Nous considérons le cas où une caméra calibrée évolue dans un environnement 3D et où sa pose est estimée à intervalles réguliers par un système que nous modélisons de la manière suivante :

$$\begin{cases} R_{z_k} = R \exp_{SO(3)}^\wedge (w_{R_k}) \\ T_{z_k} = T + w_{T_k} \end{cases} \quad (2.59)$$

où

- $R_{z_k} \in SO(3)$ représente une observation bruitée de l'orientation du référentiel global par rapport à celui de la caméra
- $T_{z_k} \in \mathbb{R}^3$ est une observation bruitée de la position de la caméra dans le référentiel global
- w_{R_k} et w_{T_k} sont des bruits blancs gaussiens potentiellement corrélés

Définition de l'état X et de son groupe de Lie G Nous considérons le groupe de Lie suivant :

$$G = SO(3) \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \quad (2.60)$$

sur lequel évolue l'état X . Remarquons que nous aurions pu choisir à la place le groupe de Lie $SE(3) \times \mathbb{R}^6$ que nous utilisons dans le chapitre suivant. Mais ici nous avons préféré choisir un groupe de Lie permettant de faciliter la

compréhension des étapes clés du CD-LG-EKF. Un élément de G est représenté de manière symbolique par :

$$X = \begin{bmatrix} R & & & & \\ & Id_{3 \times 3} & \omega & & \\ & & 1 & & \\ & & & Id_{3 \times 3} & T \\ & & & & 1 \\ & & & & & Id_{3 \times 3} & v \\ & & & & & & 1 \end{bmatrix} = \begin{pmatrix} R \\ \omega \\ T \\ v \end{pmatrix}_G \quad (2.61)$$

$$X^{-1} = \begin{pmatrix} R^T \\ -\omega \\ -T \\ -v \end{pmatrix}_G \quad \text{et} \quad X_1 X_2 = \begin{pmatrix} R_1 R_2 \\ \omega_1 + \omega_2 \\ T_1 + T_2 \\ v_1 + v_2 \end{pmatrix}_G \quad (2.62)$$

L'algèbre de Lie \mathfrak{g} associée à G est donc :

$$\mathfrak{g} = \mathfrak{so}(3) \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \quad (2.63)$$

et nous avons les propriétés suivantes ($a \in \mathbb{R}^{12}$) :

$$a = [a_R^T, a_\omega^T, a_T^T, a_v^T]^T \quad (2.64)$$

$$[a]_G^\wedge = \begin{bmatrix} [a_R]_{SO(3)}^\wedge & & & & \\ & \mathbf{0}_{3 \times 3} & a_\omega & & \\ & & 0 & & \\ & & & \mathbf{0}_{3 \times 3} & a_T \\ & & & & 0 \\ & & & & & \mathbf{0}_{3 \times 3} & a_v \\ & & & & & & 0 \end{bmatrix} = \begin{pmatrix} [a_R]_{SO(3)}^\wedge \\ a_\omega \\ a_T \\ a_v \end{pmatrix}_{\mathfrak{g}} \quad (2.65)$$

$$\exp_G^\wedge(a) = \begin{pmatrix} \exp_{SO(3)}^\wedge(a_R) \\ a_\omega \\ a_T \\ a_v \end{pmatrix}_G \quad (2.66)$$

$$\text{ad}_G(a) = \begin{bmatrix} \text{ad}_{SO(3)}(a_R) & \mathbf{0}_{9 \times 9} \\ \mathbf{0}_{9 \times 9} & \mathbf{0}_{9 \times 9} \end{bmatrix} \quad (2.67)$$

puisque $SO(3)$ est la seule composante non-commutative de G .

$$\Phi_G(a) = \begin{bmatrix} \Phi_{SO(3)}(a_R) & \mathbf{0}_{9 \times 9} \\ \mathbf{0}_{9 \times 9} & \mathbf{0}_{9 \times 9} \end{bmatrix} \quad (2.68)$$

Étape de propagation Étant donné la définition de G , le modèle de mouvement (2.58) peut être réécrit sous la forme du système que nous avons considéré dans l'équation (2.3) avec

$$\Omega(X) = [\omega^T, \mathbf{0}_{1 \times 3}, v^T, \mathbf{0}_{1 \times 3}]^T \quad (2.69)$$

et

$$n = [n_R^T, n_\omega^T, n_T^T, n_v^T]^T \quad (2.70)$$

Afin d'implémenter l'étape de propagation, le seul calcul que nous devons effectuer est celui de $F(t)$:

$$F(t) = \frac{d}{d\delta} \Omega(\mu(t) \exp_G^\wedge(\delta)) \Big|_{\delta=0} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & Id_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & Id_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (2.71)$$

Définition des observations z_k et de leur groupe de Lie G' Nous considérons le groupe de Lie suivant :

$$G' = SO(3) \times \mathbb{R}^3 \quad (2.72)$$

sur lequel les observations z_k évoluent. z_k est représenté symboliquement de la manière suivante :

$$z_k = \begin{bmatrix} R_{z_k} & & \\ & Id_{3 \times 3} & T_{z_k} \\ & & 1 \end{bmatrix} = \begin{pmatrix} R_{z_k} \\ T_{z_k} \end{pmatrix}_{G'} \quad (2.73)$$

$$z_k^{-1} = \begin{pmatrix} R_{z_k}^T \\ -T_{z_k} \end{pmatrix}_{G'} \quad \text{et} \quad z_1 z_2 = \begin{pmatrix} R_{z_1} R_{z_2} \\ T_{z_1} + T_{z_2} \end{pmatrix}_{G'} \quad (2.74)$$

L'algèbre de Lie \mathfrak{g}' associée à G' est donc :

$$\mathfrak{g}' = \mathfrak{so}(3) \times \mathbb{R}^3 \quad (2.75)$$

et nous avons les propriétés suivantes ($b \in \mathbb{R}^6$) :

$$b = [b_R^T, b_T^T]^T \quad (2.76)$$

$$[b]_G^\wedge = \begin{bmatrix} [b_R]_{SO(3)}^\wedge & & \\ & \mathbf{0}_{3 \times 3} & b_T \\ & & 0 \end{bmatrix} = \begin{pmatrix} [b_R]_{SO(3)}^\wedge \\ b_T \end{pmatrix}_{\mathfrak{g}} \quad (2.77)$$

$$\exp_{G'}^\wedge(b) = \begin{pmatrix} \exp_{SO(3)}^\wedge(b_R) \\ b_T \end{pmatrix}_{G'} \quad (2.78)$$

Étape de mise à jour Étant donné la définition de G' , le modèle d'observation (2.59) peut être réécrit sous la forme du modèle d'observation que nous avons considéré dans l'équation (2.5) avec

$$h(X_k) = AX_k A^T = \begin{bmatrix} R_k & & \\ & Id_{3 \times 3} & T_k \\ & & 1 \end{bmatrix} \quad (2.79)$$

$$A = \begin{bmatrix} Id_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & Id_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & \mathbf{0}_{1 \times 3} & 1 & \mathbf{0}_{1 \times 3} & 0 \end{bmatrix} \quad (2.80)$$

et

$$w_k = [w_{R_k}^T, w_{T_k}^T] \quad (2.81)$$

Afin d'implémenter l'étape de mise à jour, le seul calcul que nous devons effectuer est celui de H_k que nous approchons de la manière suivante :

$$\begin{aligned} H_k &= -\frac{\partial}{\partial \delta} \log_{G'}^\vee \left(h \left(\mu_{k|k-1} \exp_G^\wedge(\delta) \right)^{-1} z_k \right) \Big|_{\delta=\mathbf{0}} \\ &= \frac{\partial}{\partial \delta} \log_{G'}^\vee \left(z_k^{-1} h \left(\mu_{k|k-1} \exp_G^\wedge(\delta) \right) \right) \Big|_{\delta=\mathbf{0}} \end{aligned} \quad (2.82)$$

$$\approx \begin{bmatrix} Id_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & Id_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (2.83)$$

2.3.7.2 Simulation de trajectoires

Afin d'évaluer notre formalisme sur des données synthétiques, nous devons simuler des trajectoires de la caméra. Nous le faisons de la manière suivante :

$$X(t + \delta t) = X(t) \exp_G^\wedge(\Omega(X(t)) \delta t + \delta n) \quad (2.84)$$

où

$$\delta n \sim \mathcal{N}_{\mathbb{R}^{12}}(\delta n; \mathbf{0}_{12 \times 1}, R\delta t) \quad (2.85)$$

Pour nos simulations, nous avons choisi :

$$R = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & R_\omega & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & R_v \end{bmatrix} \quad (2.86)$$

où

$$R_\omega = \begin{bmatrix} 10^{-3} & 0 & 0 \\ 0 & 10^{-3} & 0 \\ 0 & 0 & \left(\frac{\pi}{4}\right)^2 \end{bmatrix} \text{ in } (\text{rad}/s^2)^2 \quad (2.87)$$

ce qui correspond à une rotation principale autour de l'axe vertical de la caméra, et

$$R_v = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 10^{-3} \end{bmatrix} \text{ in } (m/s^2)^2 \quad (2.88)$$

ce qui correspond à un déplacement dans le plan horizontal principalement.

2.3.7.3 Simulation d'observations

Une fois que nous avons simulé une trajectoire de la caméra, nous souhaitons créer une séquence d'observations. A partir d'une matrice de covariance Q_k , nous pouvons tirer un échantillon de $\mathcal{N}_{\mathbb{R}^6}(w_k; \mathbf{0}_{6 \times 1}, Q_k)$ et simuler une observation en utilisant (2.5). Dans le but de donner un sens physique à Q_k et de créer des corrélations réalistes entre les composantes de position et d'orientation, Q_k est obtenu de la manière suivante. Tout d'abord, la trajectoire simulée est placée dans un cube 3D composé de points 3D sur chacune de ses faces. Puis, les points 3D sont reprojétés dans le plan focal de la caméra en ajoutant un bruit blanc gaussien centré d'écart type 3 pixels. Nous utilisons alors un algorithme de Gauss-Newton pour estimer la pose de la caméra à partir des points 3D et de leurs reprojections dans le plan focal. Ensuite, dans le but d'estimer la covariance de la pose, nous effectuons une approximation de Gauss-Laplace intrinsèque (voir Chapitre 3) et la matrice de covariance obtenue est utilisée comme valeur pour Q_k .

2.3.7.4 Un filtre non linéaire continu-discret avec une contrainte extrinsèque (CD-Constr-NLF)

Dans le but de comparer les performances du CD-LG-EKF, nous développons une filtre continu-discret qui prend en compte la géométrie du groupe de Lie de manière extrinsèque. L'idée est de filtrer dans un espace euclidien de plus grande dimension puis de projeter l'estimée sur la variété comme cela a été proposé dans [Simon 2010 ; Davison 2007] pour un système à temps discret. Ici, nous adaptons l'étape de propagation pour qu'elle soit capable de gérer notre modèle de propagation à temps continu. L'étape de mise à jour reste elle inchangée.

Étape de propagation Nous définissons les applications $(\cdot)^{v_G} : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^l$ qui correspond à la vectorisation d'un élément de G et $(\cdot)^{M_G} : \mathbb{R}^l \rightarrow \mathbb{R}^{n \times n}$ qui est l'application inverse de $(\cdot)^{v_G}$. Nous définissons également $\pi(\cdot) : \mathbb{R}^{n \times n} \rightarrow G$ comme la projection orthogonale de $\mathbb{R}^{n \times n}$ sur G . Dans notre application par exemple $x = (X)^{v_G} = [R_1^T, R_2^T, R_3^T, w^T, T^T, v^T]^T \in \mathbb{R}^{18 \times 1}$ où R_i correspond à la i ème colonne de R . Nous considérons l'équation de propagation suivante dans \mathbb{R}^l :

$$\begin{aligned} dx(t) &= \left(\pi((x(t))^{M_G}) \left[\Omega(\pi((x(t))^{M_G})) \right]_G^\wedge \right)^{v_G} dt \\ &\quad + \left(\pi((x(t))^{M_G}) \sum_{i=1}^p [e_i]_G^\wedge \circ dB_i(t) \right)^{v_G} \\ &= f(x(t)) dt + g(x(t)) \circ db(t) \end{aligned} \quad (2.89)$$

où

$$f(x(t)) = \left(\pi((x(t))^{M_G}) \left[\Omega(\pi((x(t))^{M_G})) \right]_G^\wedge \right)^{v_G} \quad (2.90)$$

$db(t)$ est défini dans (2.25) et $g(x(t))$ est défini de telle manière que

$$g(x(t)) \circ db = \left(\pi((x(t))^{M_G}) \sum_{i=1}^p [e_i]_G^\wedge \circ dB_i(t) \right)^{v_G} \quad (2.91)$$

Les équations de propagation des deux premiers moments ($\mu \in \mathbb{R}^{l \times 1}$ et $P \in \text{Sym}^+ \subset \mathbb{R}^{l \times l}$) de $x(t)$ peuvent être approchées par (voir [Maybeck 1979]) :

$$\frac{d\mu(t)}{dt} \simeq f(\mu(t)) \quad (2.92)$$

et

$$\frac{dP(t)}{dt} \simeq F(t) P(t) + P(t) F(t)^T + g(\mu(t)) R g(\mu(t))^T \quad (2.93)$$

où

$$F(t) = \left. \frac{df(x)}{dx} \right|_{x=\mu(t)} \quad (2.94)$$

Après avoir propagé $\mu(t)$ et $P(t)$ de t_{k-1} à t_k , nous effectuons la projection suivante :

$$\mu_{k|k-1} = \left(\pi \left(\left(\mu_{k|k-1}^- \right)^{M_G} \right) \right)^{v_G} \quad (2.95)$$

$$P_{k|k-1} = B_k P_{k|k-1}^- B_k^T \quad (2.96)$$

où

$$B_k = \left. \frac{d \left(\pi \left((x)^{M_G} \right) \right)^{v_G}}{dx} \right|_{x=\mu_{k|k-1}^-} \quad (2.97)$$

Cette projection est équivalente à effectuer une mise à jour avec une observation parfaite (sans bruit) où l'observation correspond à une contrainte d'égalité prenant une valeur nulle pour les matrices appartenant au groupe de Lie (voir [Simon 2010]).

Étape de mise à jour Nous définissons l'opérateur $(\cdot)^{v_{G'}} : G' \subset \mathbb{R}^{m \times m} \rightarrow \mathbb{R}^k$ qui correspond à la vectorisation d'un élément de G' et $(\cdot)^{M_{G'}} : \mathbb{R}^k \rightarrow G' \subset \mathbb{R}^{m \times m}$ la projection de \mathbb{R}^k sur G' . Nous considérons l'équation d'observation suivante :

$$y_k = l(x_k, w_k) = \left(h \left((x_k)^{M_G} \right) \exp_{G'}^{\wedge} (w_k) \right)^{v_{G'}} \quad (2.98)$$

Les paramètres μ et P sont alors mis à jour en utilisant les équations classiques de mise à jour du CD-EKF :

$$K_k = P_{k|k-1} H_k^T \left(H_k P_{k|k-1} H_k^T + M_k Q_k M_k^T \right)^{-1} \quad (2.99)$$

$$m_{k|k} = K_k \left(y_k - l \left(\mu_{k|k-1}, \mathbf{0} \right) \right) \quad (2.100)$$

$$\mu_{k|k}^- = \mu_{k|k-1} + m_{k|k} \quad (2.101)$$

$$P_{k|k}^- = (Id_{l \times l} - K_k H_k) P_{k|k} \quad (2.102)$$

où

$$H_k = \left. \frac{\partial l(x, w)}{\partial x} \right|_{x=\mu_{k|k-1}, w=\mathbf{0}} \quad (2.103)$$

et

$$M_k = \left. \frac{\partial l(x, w)}{\partial w} \right|_{x=\mu_{k|k-1}, w=\mathbf{0}} \quad (2.104)$$

Finalement $\mu_{k|k}^-$ et $P_{k|k}^-$ sont projetés sur G en utilisant eq.(2.95) et (2.96).

Détails d'implémentation Lors de l'implémentation de ce filtre, il faut faire attention au fait que la matrice de covariance P est singulière. En effet, dans notre problème, P est de taille 18×18 alors que l'état n'a que 12 degrés de liberté. De ce fait, le schéma d'intégration numérique décrit dans (2.56) ne peut pas être utilisé puisque le logarithme matriciel n'est pas défini pour les matrices singulières. En lieu et place, nous utilisons une méthode de Runge Kutta d'ordre 4 en s'assurant qu'à chaque pas les valeurs propres de P restent positives. Ensuite, l'inversion matricielle dans (2.99) est remplacée par une pseudo-inverse. Finalement, afin d'améliorer la stabilité numérique du filtre, nous effectuons l'étape de projection après chaque pas du schéma d'intégration numérique.

2.3.7.5 Un filtre de Kalman sans parfum continu-discret avec une contrainte extrinsèque (CD-Constr-UKF)

Dans le cas euclidien, le fait d'utiliser la transformée sans parfum [Julier 1997] au lieu d'une technique par linéarisation permet d'obtenir le filtre de Kalman sans parfum qui produit de meilleures performances de l'EKF. Nous cherchons à comparer les performances d'un tel filtre prenant en compte la géométrie du groupe de Lie de manière extrinsèque (à l'aide d'une contrainte) avec celles du CD-LG-EKF qui est basé sur une technique de linéarisation mais qui prend en compte la géométrie du groupe de Lie de manière intrinsèque.

Étape de propagation L'étape de propagation est la même que celle décrite dans [Singer 2006] puisque leur modèle permet de gérer (2.89). Cependant, à la fin de la propagation, afin d'obtenir un état appartenant au groupe de Lie, nous appliquons la même projection que celle effectuée après l'étape de propagation du CD-Constr-NLF ((2.95) et (2.96)).

Étape de mise à jour L'étape de mise à jour est la même que celle décrite dans [Singer 2006] puisque leur modèle permet de gérer (2.98). Cependant, à la fin de la mise à jour, afin d'obtenir un état appartenant au groupe de Lie, nous appliquons la même projection que celle effectuée après l'étape de mise à jour du CD-Constr-NLF ((2.95) et (2.96)).

Détails d'implémentation Comme dans le cas du CD-Constr-NLF, il convient de faire attention, lors de l'implémentation, au fait que la matrice de covariance P est singulière. Premièrement, la décomposition de Cholesky, qui est habituellement utilisée pour déterminer les sigma-points, ne peut plus l'être dans le cas d'une matrice de covariance singulière. Dès lors, nous y substituons une décomposition en valeurs singulières et prenons la racine carrée des 12

plus grandes valeurs propres. Deuxièmement, pour améliorer les performances du filtre, nous effectuons l'étape de projection après chaque pas de l'étape de propagation.

2.3.7.6 Résultats sur données simulées

Nous simulons des trajectoires comme décrit dans le paragraphe 2.3.7.2. Pour chaque trajectoire, nous créons une séquence d'observations comme expliqué dans le paragraphe 2.3.7.3. Un exemple de trajectoire simulée est présenté en figure 2.1. Pour chaque filtre, T et R sont initialisés en utilisant la première observation avec une faible covariance alors que ω et v sont initialisés à zéro avec une grande covariance. Le pas d'échantillonnage utilisé pour simuler une trajectoire est un dixième du pas de temps utilisé pour propager la moyenne et la covariance de chaque filtre. La figure 2.2 rapporte les racines carrées des erreurs moyennes quadratiques (RMSE) de chaque filtre par rapport au temps écoulé entre deux observations (Δt). Le RMSE est défini comme la racine carrée de la moyenne des erreurs suivantes : $\|\mu_T - T\|_2^2$ (erreur en position) et $\|\log_{SO(3)}^\vee(\mu_R^T R)\|_2^2$ (erreur en orientation).

Nous remarquons que pour de faibles valeurs de la période d'échantillonnage Δt , le CD-LG-EKF le CD-Constr-NLF et le CD-Constr-UKF produisent les mêmes performances.

Cependant, pour une valeur raisonnable de Δt telle que 25 mesures par seconde qui est la valeur standard pour une caméra, le CD-LG-EKF et le CD-Constr-UKF produisent de bien meilleures performances que le CD-Constr-NLF. Pour des valeurs plus élevées de Δt , le CD-Constr-NLF diverge alors que les deux autres filtres continuent de filtrer la pose de la caméra.

Remarquons tout de même que le CD-LG-EKF fournit des résultats légèrement meilleurs que le CD-Constr-UKF que ce soit en position ou en orientation.

Pour de plus grandes valeurs de Δt , le modèle de mouvement devient moins informatif mais le CD-LG-EKF demeure numériquement stable et tend vers le RMSE des observations. Au contraire, le CD-Constr-UKF diverge.

Nous pouvons donc déduire de ces résultats que le fait de prendre en compte la géométrie du groupe de Lie de manière intrinsèque permet d'augmenter sensiblement les performances d'un filtre euclidien avec une contrainte extrinsèque et permet même d'obtenir des performances légèrement meilleures que celles d'un filtre utilisant une transformée sans parfum.

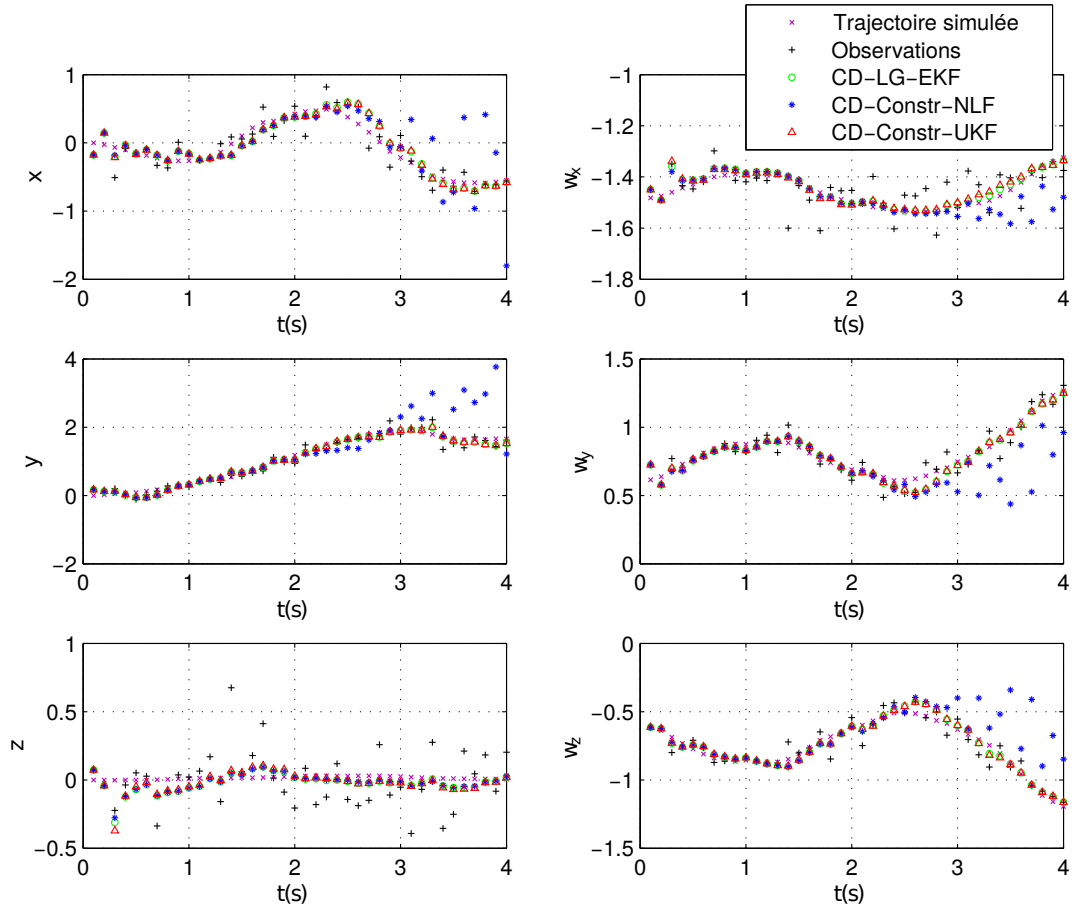


Figure 2.1 – Un exemple d’une trajectoire simulée, d’observations simulées, et du résultat obtenu avec les 3 filtres : CD-Constr-NLF, CD-Constr-UKF et CD-LG-EKF. $[x, y, z]^T = T$ et $[w_x, w_y, w_z]^T = \log_{SO(3)}^\vee(R)$.

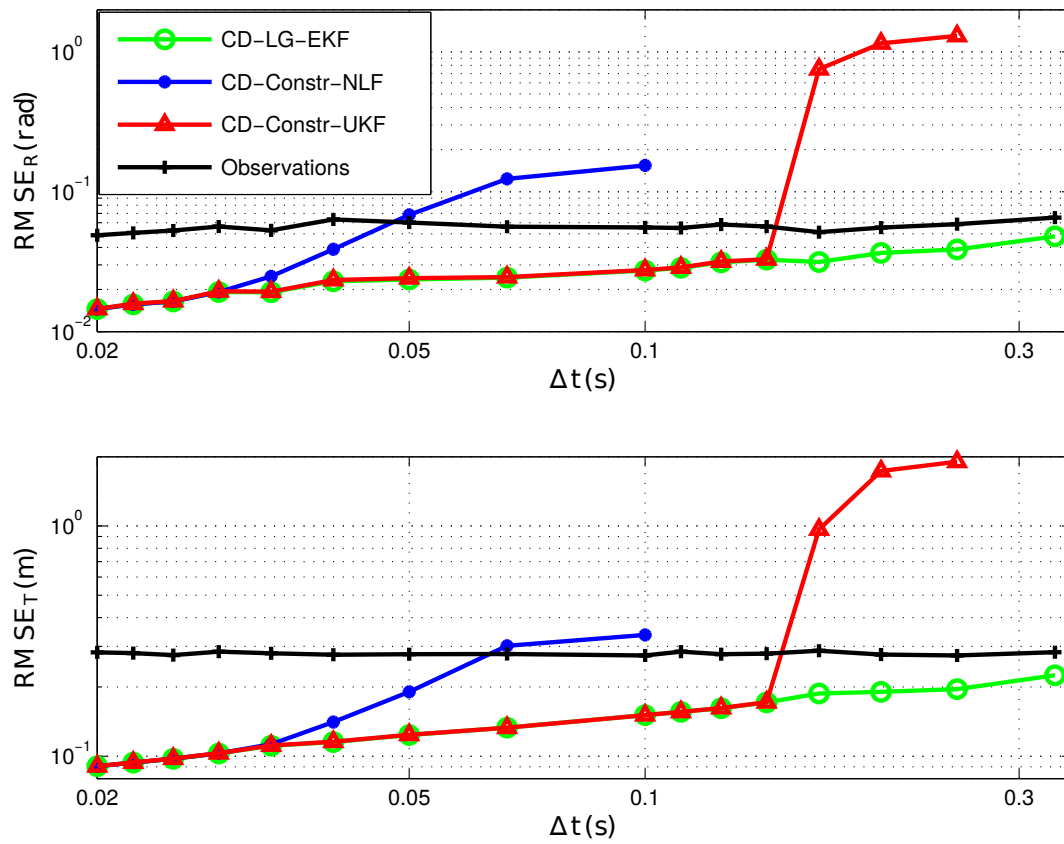


Figure 2.2 – RMSE pour différentes périodes d'échantillonnage (Δt)

2.4 Filtre de Kalman étendu discret

Dans la première partie de ce chapitre, nous avons proposé une méthode permettant de propager et mettre à jour la moyenne et la covariance d'un état dont l'évolution est décrite par une équation différentielle stochastique sur groupe de Lie et ce à partir d'observations évoluant également sur un groupe de Lie. Nous proposons ici une solution dédiée à un problème très similaire. En effet, au lieu de chercher à résoudre un problème continu-discret, nous allons tout d'abord approcher au premier ordre l'équation différentielle stochastique par un modèle de propagation à temps discret. Le problème à résoudre est alors un problème à temps discret. En partant du même principe que pour le CD-LG-EKF, c'est à dire en supposant que la distribution a posteriori de l'état est une distribution normale concentrée, nous obtenons des équations permettant de propager et mettre à jour la moyenne et la covariance de cette distribution. Nous montrons également que dans le cas où l'état et les observations évoluent sur des espaces euclidiens, le filtre obtenu, appelé filtre de Kalman étendu discret sur groupe de Lie (D-LG-EKF), se réduit au traditionnel EKF.

2.4.1 Rappels sur le filtrage de Kalman étendu discret sur espace euclidien

Nous rappelons ici brièvement l'algorithme EKF [Maybeck 1979].

2.4.1.1 Modèles de propagation et d'observation

L'algorithme EKF permet d'estimer l'état $x \in \mathbb{R}^p$ d'un système dynamique, décrit par un modèle de propagation non linéaire à temps discret, à partir d'observations discrètes reliées à l'état par une équation non linéaire :

$$\begin{cases} x_k &= f(x_{k-1}) + n_k \\ z_k &= h(x_k) + w_k \end{cases} \quad (2.105)$$

où $x_k = x(t = k)$, $z_k = z(t_k) \in \mathbb{R}^q$ est une observation reçue à l'instant $t = k$. De plus, $n_k \sim \mathcal{N}_{\mathbb{R}^p}(n_k; \mathbf{0}, R_k)$ et $w_k \sim \mathcal{N}_{\mathbb{R}^q}(w_k; \mathbf{0}, Q_k)$ sont des bruits blancs gaussiens alors que $f(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}^p$ et $h(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}^q$ sont des fonctions différentiables potentiellement non linéaires.

2.4.1.2 Principe

Une manière d'interpréter l'EKF est de supposer qu'à chaque instant la distribution a posteriori de x est une distribution normale multivariée de moyenne

μ et de covariance P . L'objectif de l'EKF est alors d'estimer les paramètres μ et P à chaque instant sachant toutes les observations précédemment reçues. L'algorithme obtenu opère en deux étapes qui alternent. Supposons que les paramètres $\mu_{k-1|k-1}$ et $P_{k-1|k-1}$ de la distribution $p(x_{k-1}|z_1, z_2, \dots, z_{k-1}) \approx \mathcal{N}_{\mathbb{R}^p}(x_{k-1}; \mu_{k-1|k-1}, P_{k-1|k-1})$ sont connus. L'étape de propagation consiste alors, comme son nom l'indique, à propager la moyenne $\mu_{k-1|k-1}$ et la covariance $P_{k-1|k-1}$ afin d'obtenir les paramètres $\mu_{k|k-1}$ et $P_{k|k-1}$ de la distribution $p(x_k|z_1, z_2, \dots, z_{k-1}) \approx \mathcal{N}_{\mathbb{R}^p}(x_k; \mu_{k|k-1}, P_{k|k-1})$. Ces deux paramètres sont alors mis à jour en utilisant l'information apportée par la nouvelle observation z_k pour obtenir les paramètres $\mu_{k|k}$ et $P_{k|k}$ de la distribution $p(x_k|z_1, z_2, \dots, z_{k-1}, z_k) \approx \mathcal{N}_{\mathbb{R}^p}(x_k; \mu_{k|k}, P_{k|k})$. Dans le cas de l'EKF, les équations permettant de propager et de mettre à jour la moyenne et la covariance sont obtenues en linéarisant les modèles de propagation et d'observation.

2.4.1.3 Algorithme EKF

L'algorithme du EKF est présenté ci-après :

Algorithme 2.3 EKF

Entrée : $\mu_{k-1|k-1}, P_{k-1|k-1}, z_k$

Sortie : $\mu_{k|k}, P_{k|k}$

Propagation :

$$\mu_{k|k-1} = f(\mu_{k-1|k-1})$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + R_k \text{ où } F_k = \left. \frac{d}{dx} f(x) \right|_{x=\mu_{k-1|k-1}}$$

Mise à jour :

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + Q_k)^{-1} \text{ où } H_k = \left. \frac{d}{dx} h(x) \right|_{x=\mu_{k|k-1}}$$

$$\tilde{z}_k = z_k - h(\mu_{k|k-1})$$

$$m_{k|k}^- = K_k \tilde{z}_k$$

$$\mu_{k|k} = \mu_{k|k-1} + m_{k|k}^-$$

$$P_{k|k} = (Id - K_k H_k) P_{k|k-1}$$

Cet algorithme, qui est considéré comme une méthode de référence dans le domaine de l'estimation non linéaire à temps discret, est restreint au cas où l'état et les observations évoluent sur des espaces euclidiens. Dans la suite de ce paragraphe, nous proposons d'étendre cette approche au cas où l'état et les observations évoluent sur des groupes de Lie.

2.4.2 Modèles de propagation et d'observation proposés

2.4.2.1 Modèle de propagation

Nous considérons l'équation de propagation suivante qui est une approximation au premier ordre de l'équation de propagation du CD-LG-EKF (voir paragraphe 2.3.2.1) :

$$X_k = X_{k-1} \exp_G^\wedge (\Omega(X_{k-1}) + n_k) \quad (2.106)$$

où $X_k \in G$ est l'état que nous souhaitons estimer au temps $t = t_k$ et G est un groupe de Lie matriciel de dimension p . La variable $n_k \sim \mathcal{N}_{\mathbb{R}^p}(n_k; \mathbf{0}_{p \times 1}, R_k)$ est un bruit blanc gaussien. La fonction $\Omega : G \rightarrow \mathbb{R}^p$ est différentiable et potentiellement non linéaire.

Remarquons que nous aurions pu choisir un modèle de propagation plus générique comme nous le verrons dans le chapitre 3. Nous souhaitons ici proposer une alternative à temps discret au CD-LG-EKF, c'est pourquoi le modèle de propagation (2.106) a été choisi comme étant une approximation au 1er ordre du modèle de propagation du CD-LG-EKF (2.2).

2.4.2.2 Modèle d'observation

Le modèle d'observation que nous considérons est le même que celui du CD-LG-EKF (voir paragraphe 2.3.2.2).

2.4.3 Solution proposée

Afin d'estimer l'état X , nous proposons de supposer que sa distribution a posteriori a la forme d'une distribution normale concentrée (à gauche) :

$$p(X_k | z_1, \dots, z_l) \approx \mathcal{N}_G^L(X_k; \mu_{k|l}, P_{k|l}) \quad (2.107)$$

Plus précisément, nous nous intéressons aux cas où $l = k - 1$ (étape de propagation) et $l = k$ (étape de mise à jour). Ainsi, notre objectif est de montrer comment propager et mettre à jour les paramètres μ et P de la distribution normale concentrée. Dans notre formalisme, nous choisissons comme estimateur de l'état X la "moyenne" μ de la distribution normale concentrée.

2.4.4 Propagation

Commençons par supposer que :

$$p(X_{k-1} | z_1, \dots, z_{k-1}) \approx \mathcal{N}_G^L(X_{k-1}; \mu_{k-1|k-1}, P_{k-1|k-1}) \quad (2.108)$$

où $\mu_{k-1|k-1}$ et $P_{k-1|k-1}$ sont connus, et

$$p(X_k|z_1, \dots, z_{k-1}) \approx \mathcal{N}_G^L(X_k; \mu_{k|k-1}, P_{k|k-1}) \quad (2.109)$$

où $\mu_{k|k-1}$ et $P_{k|k-1}$ sont inconnus. L'objectif de l'étape de propagation est de montrer comment propager $\mu_{k-1|k-1}$ et $P_{k-1|k-1}$ en utilisant le modèle de propagation (2.106) afin d'obtenir $\mu_{k|k-1}$ et $P_{k|k-1}$.

Nous rappelons également que puisque (2.108) a la forme d'une distribution normale concentrée à gauche, X_{k-1} peut alors s'exprimer de la manière suivante :

$$X_{k-1} = \mu_{k-1|k-1} \exp_G^\wedge(\epsilon_{k-1}) \quad (2.110)$$

où l'erreur dans l'algèbre de Lie ϵ_{k-1} est distribuée de la manière suivante :

$$\epsilon_{k-1} \sim \mathcal{N}_{\mathbb{R}^p}(\epsilon_{k-1}; m_{k-1|k-1} = \mathbf{0}, P_{k-1|k-1}) \quad (2.111)$$

2.4.4.1 Propagation de la moyenne

Nous choisissons de propager la moyenne μ en utilisant le modèle d'état (2.106) sans bruit :

$$\mu_{k|k-1} = \mu_{k-1|k-1} \exp_G^\wedge(\Omega(\mu_{k-1|k-1})) \quad (2.112)$$

Nous montrons par la suite qu'en propageant la moyenne de cette manière, la moyenne de l'erreur dans l'algèbre de Lie $m_{k|k-1} = \mathbb{E}[\epsilon_k]$ demeure nulle au premier ordre.

2.4.4.2 Propagation de la covariance

Afin de trouver une expression pour la propagation de la covariance, nous étudions la propagation de l'erreur dans l'algèbre de Lie où nous utilisons (2.112) et (2.106) :

$$\begin{aligned} \exp_G^\wedge(\epsilon_k) &= \mu_{k|k-1}^{-1} X_k \\ &= \exp_G^\wedge(-\Omega(\mu_{k-1|k-1})) \mu_{k-1|k-1}^{-1} X_{k-1} \exp_G^\wedge(\Omega(X_{k-1}) + n_k) \\ &= \exp_G^\wedge(-\Omega(\mu_{k-1|k-1})) \exp_G^\wedge(\epsilon_{k-1}) \exp_G^\wedge(\Omega(X_{k-1}) + n_k) \\ &= \exp_G^\wedge(\text{Ad}_G(\exp_G^\wedge(-\Omega(\mu_{k-1|k-1}))) \epsilon_{k-1}) \\ &\quad \exp_G^\wedge(-\Omega(\mu_{k-1|k-1})) \exp_G^\wedge(\Omega(X_{k-1}) + n_k) \end{aligned} \quad (2.113)$$

En linéarisant Ω en $X_{k-1} = \mu_{k-1|k-1}$, nous obtenons :

$$\Omega(X_{k-1}) = \Omega(\mu_{k-1|k-1}) + \mathcal{C}_k \epsilon_{k-1} + O(\|\epsilon_{k-1}\|^2) \quad (2.114)$$

où

$$\mathcal{C}_k = \left. \frac{\partial}{\partial \delta} \Omega \left(\mu_{k-1|k-1} \exp_G^\wedge(\delta) \right) \right|_{\delta=\mathbf{0}} \quad (2.115)$$

Ainsi en utilisant (1.29) :

$$\begin{aligned} & \exp_G^\wedge \left(-\Omega \left(\mu_{k-1|k-1} \right) \right) \exp_G^\wedge \left(\Omega \left(X_{k-1} \right) + n_k \right) \\ &= \exp_G^\wedge \left(-\Omega \left(\mu_{k-1|k-1} \right) \right) \exp_G^\wedge \left(\Omega \left(\mu_{k-1|k-1} \right) + \mathcal{C}_k \epsilon_{k-1} + O \left(\|\epsilon_{k-1}\|^2 \right) + n_k \right) \\ &= \exp_G^\wedge \left(\Phi_G \left(-\Omega \left(\mu_{k-1|k-1} \right) \right) \left(\mathcal{C}_k \epsilon_{k-1} + n_k \right) + O \left(\|\epsilon_{k-1}, n_k\|^2 \right) \right) \end{aligned} \quad (2.116)$$

En reportant ce résultat dans (2.113), nous obtenons finalement :

$$\epsilon_k = F_k \epsilon_{k-1} + \Phi_G \left(-\Omega \left(\mu_{k-1|k-1} \right) \right) n_k + O \left(\|\epsilon_{k-1}, n_k\|^2 \right)$$

où

$$F_k = \text{Ad}_G \left(\exp_G^\wedge \left(-\Omega \left(\mu_{k-1|k-1} \right) \right) \right) + \Phi_G \left(-\Omega \left(\mu_{k-1|k-1} \right) \right) \mathcal{C}_k \quad (2.117)$$

En négligeant les termes en $O \left(\|\epsilon_{k-1}, n_k\|^2 \right)$, nous avons donc bien que $m_{k|k-1} = \mathbb{E}[\epsilon_k] = \mathbf{0}_{p \times 1}$. L'équation de propagation de la covariance s'écrit alors :

$$\begin{aligned} P_{k|k-1} &= \mathbb{E} \left[\epsilon_k \epsilon_k^T \right] \\ &= F_k P_{k-1|k-1} F_k^T + \Phi_G \left(-\Omega \left(\mu_{k-1|k-1} \right) \right) R_k \Phi_G \left(-\Omega \left(\mu_{k-1|k-1} \right) \right)^T \end{aligned} \quad (2.118)$$

2.4.4.3 Résumé de l'étape de propagation

L'étape de propagation consiste, comme son nom l'indique, à propager la moyenne $\mu_{k-1|k-1}$ et la covariance $P_{k-1|k-1}$ en utilisant le modèle de propagation (2.106) afin d'obtenir la moyenne $\mu_{k|k-1}$ et la covariance $P_{k|k-1}$. A la fin de l'étape de propagation la distribution a posteriori de l'état est alors paramétrée de la manière suivante :

$$p(X_k | z_1, \dots, z_{k-1}) \approx \mathcal{N}_G^L \left(X_k; \mu_{k|k-1}, P_{k|k-1} \right) \quad (2.119)$$

2.4.5 Mise à jour

Le modèle d'observation considéré étant le même que celui du CD-LG-EKF, et la distribution a posteriori $p(X_k | z_1, \dots, z_{k-1})$ à l'issue de la propagation étant paramétrée de la même manière que dans le cas du CD-LG-EKF, l'étape de mise à jour est par conséquent identique à celle du CD-LG-EKF (voir paragraphe (2.3.5)).

2.4.6 Algorithme D-LG-EKF

L'algorithme formé par les étapes de propagation et de mise à jour présentées précédemment est appelé filtre de Kalman étendu discret sur groupe de Lie (D-LG-EKF).

2.4.6.1 Vue générale

L'algorithme D-LG-EKF est résumé ci-après (voir paragraphe 2.4.4 et paragraphe 2.4.5 pour plus de détails) :

Algorithme 2.4 D-LG-EKF

Entrée : $\mu_{k-1|k-1}$, $P_{k-1|k-1}$, z_k

Sortie : $\mu_{k|k}$, $P_{k|k}$

Propagation :

$$\begin{aligned}\mu_{k|k-1} &= \mu_{k-1|k-1} \exp_G^\wedge \left(\Omega \left(\mu_{k-1|k-1} \right) \right) \\ P_{k|k-1} &= F_k P_{k-1|k-1} F_k^T + \Phi_G \left(-\Omega \left(\mu_{k-1|k-1} \right) \right) R_k \Phi_G \left(-\Omega \left(\mu_{k-1|k-1} \right) \right)^T\end{aligned}$$

Mise à jour :

$$\begin{aligned}K_k &= P_{k|k-1} H_k^T \left(H_k P_{k|k-1} H_k^T + Q_k \right)^{-1} \\ \tilde{z}_k &= \log_G^\vee \left(h \left(\mu_{k|k-1} \right)^{-1} z_k \right) \\ m_{k|k}^- &= K_k \tilde{z}_k \\ \mu_{k|k} &= \mu_{k|k-1} \exp_G^\wedge \left(m_{k|k}^- \right) \\ P_{k|k} &= \Phi_G \left(-m_{k|k}^- \right) (Id - K_k H_k) P_{k|k-1} \Phi_G \left(-m_{k|k}^- \right)^T\end{aligned}$$

2.4.6.2 Une généralisation de l'EKF

Dans le cas spécifique où G , le groupe de Lie sur lequel évolue l'état X , et G' , le groupe de Lie sur lequel évolue les observations, sont des espaces euclidiens, alors l'algorithme D-LG-EKF prend la forme d'un EKF. En effet, le modèle de propagation devient une fonction non linéaire de l'état X_{k-1} avec un bruit additif, et l'équation d'observation prend la forme d'une fonction non linéaire de l'état X_k avec un bruit additif. Le D-LG-EKF peut donc être vu comme une généralisation de l'EKF.

2.4.7 Application du D-LG-EKF à un problème de filtrage de la trajectoire d'une caméra

Comme dans le cas du CD-LG-EKF, nous nous intéressons au problème de l'estimation de la pose d'une caméra. Plus précisément, nous considérons

un modèle de propagation à temps discret où l'accélération est un bruit blanc et un modèle de mesure où l'observation correspond à la pose de la caméra bruitée.

2.4.7.1 Obtention des équations du filtre

Modèle de mouvement Nous considérons le modèle de mouvement suivant qui est une approximation au premier ordre du modèle de propagation (2.58) :

$$\begin{cases} R_k = R_{k-1} \exp_{SO(3)}^{\wedge} (\omega_{k-1} \Delta t + n_R) \\ \omega_k = \omega_{k-1} + n_\omega \\ T_k = T_{k-1} + v_{k-1} \Delta t + n_T \\ v_k = v_{k-1} + n_v \end{cases} \quad (2.120)$$

où

- $R = R_{gb} \in SO(3)$, représente l'orientation du référentiel global (g) par rapport au référentiel de la caméra (b)
- $\omega = \omega_b \in \mathbb{R}^3$ est la vitesse angulaire dans le référentiel de la caméra
- $T = \vec{O_g O_b}^g \in \mathbb{R}^3$ est la position de la caméra dans le référentiel global
- $v = v_g \in \mathbb{R}^3$ est la vitesse radiale dans le référentiel global
- n_R, n_ω, n_T et n_v sont des bruits blancs gaussiens (potentiellement corrélés)

Modèle d'observation Le modèle d'observation utilisé est le même que (2.59).

Définition de l'état X et de son groupe de Lie G L'état X et le groupe de Lie G sont les mêmes que pour l'application du CD-LG-EKF .

Étape de propagation Étant donné la définition de G , le modèle de mouvement (2.120) peut être réécrit de la forme du système que nous avons considéré dans l'équation (2.106) avec

$$\Omega(X) = [\omega^T \Delta t, \mathbf{0}_{1 \times 3}, v^T \Delta t, \mathbf{0}_{1 \times 3}]^T \quad (2.121)$$

et

$$n = [n_R^T, n_\omega^T, n_T^T, n_v^T]^T \quad (2.122)$$

Afin d'implémenter l'étape de propagation, le seul calcul que nous devons effectuer est celui de \mathcal{C}_k dont l'expression est utilisée par F_k (2.117) :

$$\mathcal{C}_k = \frac{\partial}{\partial \delta} \Omega \left(\mu_{k-1|k-1} \exp_G^\wedge(\delta) \right) \Big|_{\delta=0} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \Delta t \cdot Id_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \Delta t \cdot Id_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (2.123)$$

Définition des observations z_k et de leur groupe de Lie G' Les observations z_k et le groupe de Lie G' sont les mêmes que pour l'application du CD-LG-EKF .

Étape de mise à jour Cette étape est la même que pour l'application du CD-LG-EKF .

2.4.7.2 Simulation de trajectoires

Afin d'évaluer notre formalisme sur des données synthétiques, nous simulons des trajectoires de la caméra en utilisant le modèle de propagation (2.120). Pour nos simulations, nous avons arbitrairement choisi les valeurs suivantes en nous inspirant du raisonnement proposé dans [Maybeck 1979] :

$$R = \begin{bmatrix} R_\omega & \mathbf{0}_{6 \times 6} \\ \mathbf{0}_{6 \times 6} & R_v \end{bmatrix} \quad (2.124)$$

avec

$$R_\omega = 10^{-12} \left(A\Delta t + BA\frac{\Delta t^2}{2} + AB^T\frac{\Delta t^2}{2} + BAB^T\frac{\Delta t^3}{3} \right) \quad (2.125)$$

et

$$R_v = 10^{-7} \left(A\Delta t + BA\frac{\Delta t^2}{2} + AB^T\frac{\Delta t^2}{2} + BAB^T\frac{\Delta t^3}{3} \right) \quad (2.126)$$

où

$$A = \begin{bmatrix} \mathbf{0}_{3 \times 3} & Id \end{bmatrix}^T \begin{bmatrix} \mathbf{0}_{3 \times 3} & Id \end{bmatrix} \quad (2.127)$$

et

$$B = \begin{bmatrix} \mathbf{0}_{3 \times 3} & Id \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (2.128)$$

2.4.7.3 Simulation d'observations

Le modèle d'observation étant identique à celui du CD-LG-EKF, les séquences d'observations sont obtenues de la même manière que pour l'application du CD-LG-EKF.

2.4.7.4 Filtres euclidiens

Nous avons choisi de comparer les performances du D-LG-EKF par rapport à deux filtres EKF. Le premier, que nous appelons EKF-Constr, est un filtre de Kalman étendu contraint [Simon 2010] qui est donc une version à temps discret du CD-Constr-NLF (voir paragraphe (2.3.7.4)). Le second est un EKF qui utilise une paramétrisation de G . Il tire parti du fait que pour $SO(3)$ le logarithme matriciel est défini pour tous les éléments du groupe. Nous appelons ce filtre EKF-LieAlg.

2.4.7.5 Résultats sur données simulées

Nous simulons des trajectoires comme décrit dans le paragraphe 2.4.7.2. Pour chaque trajectoire, nous créons une séquence d'observations comme expliqué dans le paragraphe 2.4.7.3. Pour chaque filtre, T et R sont initialisés en utilisant la première observation avec une faible covariance alors que ω et v sont initialisés à zéro avec une grande covariance. La figure 2.3 rapporte les racines carrées des erreurs moyennes quadratiques (RMSE) de chaque filtre par rapport au temps écoulé entre deux observations (Δt). Le RMSE est défini comme la racine carrée de la moyenne des erreurs suivantes : $\|\mu_T - T\|_2^2$ (erreur en position) et $\|\log_{SO(3)}^\vee(\mu_R^T R)\|_2^2$ (erreur en orientation).

Remarquons tout d'abord que l'EKF-LieAlg n'est pas capable de filtrer correctement. En effet, la paramétrisation utilisée conduit la partie "rotation" de l'état à être paramétrée par 3 paramètres formant un vecteur dont la direction peut être interprétée comme l'axe de la rotation autour duquel tourne la caméra et la norme comme l'angle de rotation. Donc deux vecteurs ayant la même direction, l'un ayant une norme proche de 2π et l'autre proche 0 représentent "presque" la même rotation. Or la distance euclidienne utilisée par l'EKF ne prend pas en compte cette périodicité, ce qui cause la divergence du filtre dès qu'il se trouve dans cette situation.

Nous remarquons que pour de faibles valeurs de la période d'échantillonnage Δt , le D-LG-EKF et l'EKF-Constr produisent les mêmes performances. Cependant, plus Δt augmente plus le RMSE de l'EKF-Constr se dégrade alors que le D-LG-EKF continue de fournir de bonnes performances. Pour des valeurs plus élevées de Δt , l'EKF-Constr diverge alors que le D-LG-EKF continue de filtrer la pose de la caméra.

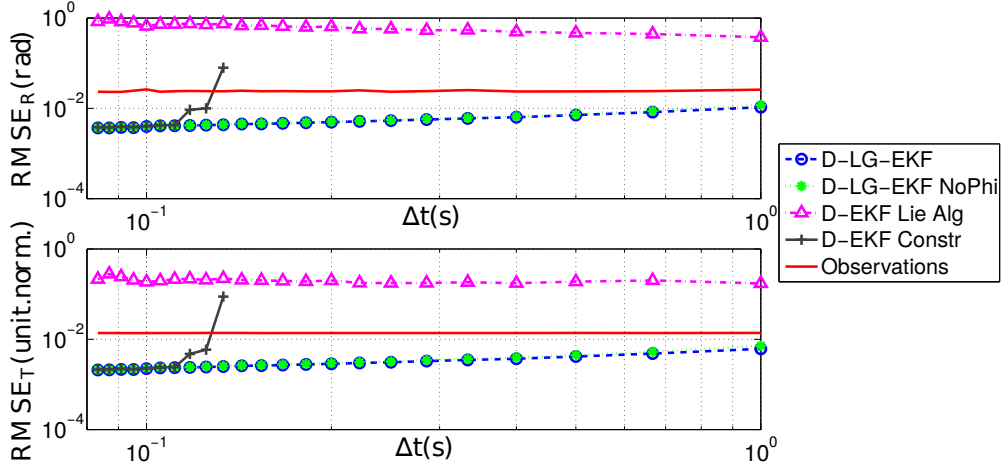


Figure 2.3 – RMSE pour différentes périodes d’échantillonnage (Δt)

Finalement, nous nous sommes également intéressés au cas où l’on remplace les matrices Φ_G par des matrices identité dans l’implémentation du D-LG-EKF. Nous appelons cette version de l’algorithme D-LG-EKF-NoPhi. Il en ressort que le D-LG-EKF-NoPhi produit des performances presque aussi satisfaisantes que le D-LG-EKF pour un coût calculatoire réduit.

Nous n’avons pas comparé les performances du CD-LG-EKF par rapport à celles du D-LG-EKF puisqu’ils n’ont pas le même modèle de propagation. Le modèle de propagation du D-LG-EKF est simplement une approximation au 1er ordre de celui du CD-LG-EKF. Cependant, en pratique, nous pouvons nous attendre à obtenir des performances similaires car les deux filtres se sont montrés stables numériquement sur des simulations avec des paramètres semblables. De plus, notre schéma numérique d’intégration permettant de propager la moyenne et la covariance du CD-LG-EKF est une méthode du 1er ordre, ce qui est donc “presque” équivalent à l’exécution de multiples petites propagations consécutives du D-LG-EKF dont le modèle de propagation est une approximation au 1er ordre de celui du CD-LG-EKF.

2.5 Résumé des contributions et conclusion

Dans ce chapitre, nous nous sommes intéressés au problème du filtrage de Kalman à temps discret et continu-discret à partir d’observations appartenant à un groupe de Lie, où l’état évolue également sur une groupe de Lie. Nous avons proposé deux filtres ; le CD-LG-EKF qui permet de résoudre un problème à temps continu-discret et le D-LG-EKF qui permet de résoudre un problème à temps discret.

Ces deux filtres sont basés sur la même approximation, à savoir qu’à chaque

instant, nous supposons que la distribution a posteriori de l'état est une distribution normale concentrée sur groupe de Lie.

En utilisant le formalisme des groupes de Lie, nous avons montré comment propager et mettre à jour les paramètres de cette distribution.

Nous avons également montré que le CD-LG-EKF peut être vu comme une généralisation du CD-EKF alors que le D-LG-EKF se réduit à l'EKF lorsque l'état et les observations évoluent sur des espaces euclidiens.

Finalement, nous avons comparé les performances de ces deux filtres sur une application de filtrage de la trajectoire d'une caméra. Et dans les deux cas, les filtres proposés se sont montrés numériquement stables, et ont produit de meilleures performances que les algorithmes de l'état de l'art du point de vue du RMSE.

Chapitre 3

De l'optimisation intrinsèque au filtrage et lissage de Kalman sur groupes de Lie matriciels

3.1	Introduction	85
3.2	État de l'art	86
3.3	Approximation de Gauss-Laplace intrinsèque	87
3.3.1	Problème	87
3.3.2	Solution proposée	87
3.4	Algorithme de Gauss-Newton intrinsèque	89
3.4.1	Introduction à l'optimisation intrinsèque	89
3.4.2	Algorithme LG-GN	90
3.5	Filtre de Kalman étendu itéré	91
3.5.1	Rappels sur le filtrage de Kalman étendu itéré sur espace euclidien	91
3.5.2	Modèles de propagation et d'observation proposés	94
3.5.3	Solution proposée	96
3.5.4	Propagation	96
3.5.5	Mise à jour	98
3.5.6	Algorithme LG-IEKF	102
3.5.7	Application du LG-IEKF au débruitage de transformations relatives	105
3.6	Lisseur de Rauch–Tung–Striebel	120
3.6.1	Rappels sur le lisseur de Rauch–Tung–Striebel sur espace euclidien	120
3.6.2	Modèle proposé	122
3.6.3	Solution proposée	123
3.6.4	Application du LG-GN	124
3.6.5	Algorithme LG-RTS	127
3.6.6	Application du LG-RTS sur un problème de lissage de la trajectoire d'une caméra	129
3.7	Résumé des contributions et conclusion	131

3.1 Introduction

Durant ces quinze dernières années, de nombreux algorithmes d'optimisation, permettant l'estimation de paramètres évoluant sur des variétés, ont été proposés [Absil 2009] et employés dans diverses applications en traitement du signal, en robotique ou encore en vision par ordinateur [Lui 2011].

Cependant, lorsque les paramètres sont régis par un système dynamique et/ou lorsque les observations sont acquises séquentiellement, il est important de pouvoir les estimer récursivement en utilisant un algorithme de filtrage ou de lissage [Haykin 2001].

Le lien entre l'optimisation et le filtrage/lissage bayésien, pour des paramètres euclidiens et dans le contexte des moindres carrés non linéaires, a déjà été étudié [Bell 1993; Bertsekas 1996]. Cette analyse a donné lieu au filtre de Kalman itéré étendu (IEKF), qui produit en pratique de meilleurs résultats que le simple EKF, ainsi qu'au lisseur de Kalman étendu également connu sous le nom de lisseur de Rauch-Tung-Striebel (RTS).

Bien que plusieurs travaux aient proposé des généralisations des algorithmes de filtrage euclidien aux variétés (voir Tableau 2.1), à notre connaissance, le lien entre optimisation et filtrage/lissage bayésien sur variété n'a pas encore été étudié.

Dans ce chapitre, nous proposons d'établir cette connexion pour des paramètres évoluant sur des groupes de Lie. Plus particulièrement, nous nous intéressons à l'équivalence que l'on peut établir entre la formulation de moindres carrés non linéaires et le filtrage/lissage de Kalman sur groupe de Lie.

Dans une première partie, nous proposons une technique d'ajustement, appelée approximation de Gauss-Laplace intrinsèque, permettant d'approcher la densité de probabilité d'une variable aléatoire vivant sur un groupe de Lie par une distribution normale concentrée sur groupe de Lie. Cette technique d'ajustement nécessitant de trouver le mode de la distribution cible, nous présentons un algorithme d'optimisation de type Gauss-Newton permettant de prendre en compte intrinsèquement la structure du groupe de Lie sur lequel vit la variable aléatoire. Les concepts et méthodes présentés dans cette partie seront utilisés tout au long du chapitre.

Dans une seconde partie, nous montrons qu'en utilisant l'approximation de Gauss-Laplace intrinsèque il est possible d'obtenir une généralisation de l'IEKF au cas où l'état et les observations évoluent sur des groupes de Lie. Cette technique d'ajustement conduit notamment à la formulation de critères de moindres carrés non linéaires. Nous démontrons qu'en minimisant ces critères à l'aide d'un algorithme de Gauss-Newton sur groupe de Lie il est possible d'obtenir les équations d'un nouveau filtre, que nous appelons filtre de Kalman

itéré étendu sur groupe de Lie (LG-IEKF), établissant ainsi un lien entre optimisation et filtrage bayésien sur groupe de Lie. De plus, l'algorithme obtenu, qui permet de résoudre un problème de filtrage à temps discret, est comparé à l'algorithme D-LG-EKF présenté dans le chapitre précédent.

Dans une troisième partie, nous proposons, d'une manière similaire, une généralisation du lisseur RTS aux groupes de Lie.

3.2 État de l'art

Un nombre important de travaux formule des critères de moindres carrés non linéaires afin d'estimer des paramètres évoluant sur des groupes de Lie. La plupart d'entre eux emploient une version modifiée de l'algorithme de Gauss-Newton (GN) [Björck 1996] ou de Levenberg-Marquardt afin de prendre en compte la structure du groupe de Lie sur lequel évoluent les paramètres. [Taylor 1994] fut un des premiers à proposer une modification du GN pour estimer une orientation 3D. Dans un contexte différent, [Li 2009] utilise un algorithme similaire à celui de [Taylor 1994] pour estimer des homographies. Dans le domaine de la cartographie et de la localisation simultanée (SLAM), diverses approches [Grisetti 2012; Jeong 2010; Klein 2007; Konolige 2010] emploient un algorithme semblable au GN dans le but d'estimer des points 3D ainsi que la pose de la caméra à partir d'une séquence vidéo. Dans le contexte spécifique du débruitage de transformations relatives, différentes modifications du GN permettant de prendre en compte le fait que les observations évoluent elles mêmes sur un groupe de Lie ont été proposées : [Chatterjee 2013] et [Govindu 2004] abordent le problème dit de la synchronisation de rotations. [Roberts 2011], de son côté, considère le problème de l'alignement de poses de caméra, [Strasdat 2011] propose une solution au problème de l'assemblage de morceaux de modèles 3D et [Meidow 2011] s'intéresse au problème de la construction d'une mosaïque d'images.

En ce qui concerne le filtrage sur variété, divers travaux ont proposé des algorithmes inspirés du filtre de Kalman dans le but d'estimer un état évoluant sur groupe de Lie. [Crassidis 2003; Markley 2003; Lefferts 1982; Persson 2012] considèrent l'estimation d'une orientation 3D. [Malis 2009] s'intéresse à l'estimation dynamique d'homographies alors que [Davison 2007] aborde le problème du SLAM.

A notre connaissance seuls quelques travaux spécifiquement dédiés au problème du SLAM ont proposé des méthodes reliant optimisation sur groupe de Lie et filtrage dans le but d'obtenir des algorithmes efficaces du point de vue calculatoire. Dans ce contexte, [Strasdat 2010b] propose un filtre d'information alors que [Kaess 2008; Kaess 2012] ont obtenu des filtres fondés sur la racine

carrée de la matrice d'information. Cependant, aucune de ces approches n'est capable de prendre en compte des observations évoluant sur un groupe de Lie ni de lisser les paramètres.

3.3 Approximation de Gauss-Laplace intrinsèque

Une façon d'aborder un problème de filtrage/lissage bayésien consiste à d'essayer d'approcher, à chaque pas de temps, la distribution a posteriori des paramètres par une distribution paramétrique.

Dans ce sous-chapitre, nous proposons une technique d'ajustement, qui sera utilisée dans le reste du chapitre, pour approcher, à chaque pas de temps, la distribution a posteriori des paramètres que nous souhaitons estimer par une distribution normale concentrée sur groupe de Lie.

3.3.1 Problème

Considérons la distribution d'une variable $X \in G$, où G est un groupe de Lie matriciel de dimension p , de la forme :

$$p(X) = \alpha e^{-\|\phi(X)\|_{\Sigma}^2} \quad (3.1)$$

où la fonction $\phi : G \rightarrow \mathbb{R}^m$ est supposée différentiable et $p < m$.

Nous souhaitons trouver une technique d'ajustement permettant d'approcher $p(X)$ par une distribution normale concentrée sur groupe de Lie.

3.3.2 Solution proposée

Tout d'abord, définissons le minimiseur de la fonction coût $\|\phi(X)\|_{\Sigma}^2$:

$$\hat{X} = \operatorname{argmin}_{X \in G} \|\phi(X)\|_{\Sigma}^2 \quad (3.2)$$

Un développement de Taylor sur groupe de Lie de ϕ au premier ordre autour de \hat{X} nous donne :

$$\phi(X) = \phi(\exp_G^{\wedge}(\delta) \hat{X}) \approx \phi(\hat{X}) + J\delta \quad (3.3)$$

où

$$J = \left. \frac{d\phi(\exp_G^{\wedge}(s) \hat{X})}{ds} \right|_{s=0} \quad (3.4)$$

et

$$\delta = \log_G^{\vee}(X \hat{X}^{-1}) \quad (3.5)$$

Nous supposons ici que le rang de J est égal à p . Nous choisissons alors comme approximation de $p(X)$ la distribution $q(X)$:

$$q(X) = \beta e^{-\|\phi(\hat{X}) + J \log_G^\vee(X \hat{X}^{-1})\|_\Sigma^2} \quad (3.6)$$

Nous allons maintenant montrer que $q(X)$ peut être mis sous la forme d'une distribution normale concentrée (à droite) sur groupe de Lie.

Rappelons que puisque \hat{X} est le minimiseur de $\|\phi(X)\|_\Sigma^2$, l'expression $\|\phi(\hat{X}) + J \log_G^\vee(X \hat{X}^{-1})\|_\Sigma^2$ atteint son minimum en $\log_G^\vee(X \hat{X}^{-1}) = 0$.

Remarquons maintenant que $\|\phi(\hat{X}) + J \log_G^\vee(X \hat{X}^{-1})\|_\Sigma^2$ est de la forme $\|Ax - b\|_\Sigma^2$ avec $A = J$, $x = \log_G^\vee(X \hat{X}^{-1})$ et $b = -\phi(\hat{X})$ qui atteint, dans notre cas, son minimum en $\hat{x} = 0$. Or il est possible de montrer que (voir annexe E) :

$$\begin{aligned} \|Ax - b\|_\Sigma^2 &= (x - \hat{x})^T P^{-1} (x - \hat{x}) + cst \\ &= x^T P^{-1} x + cst \end{aligned} \quad (3.7)$$

où $P^{-1} = A^T \Sigma^{-1} A$ et cst est un terme constant par rapport à x .

Nous avons donc montré que :

$$\|\phi(\hat{X}) + J \log_G^\vee(X \hat{X}^{-1})\|_\Sigma^2 = \log_G^\vee(X \hat{X}^{-1})^T P^{-1} \log_G^\vee(X \hat{X}^{-1}) + cst \quad (3.8)$$

Par conséquent, l'approximation $q(X)$ de $p(X)$ est de la forme :

$$q(X) \propto e^{-\|\log_G^\vee(X \hat{X}^{-1})\|_P^2} \quad (3.9)$$

ce qui correspond à une distribution normale concentrée (à droite) :

$$q(X) = \mathcal{N}_G^R(X; \hat{X}, P = (J^T \Sigma^{-1} J)^{-1}) \quad (3.10)$$

Nous appelons cette technique d'ajustement “approximation de Gauss-Laplace intrinsèque”. Le terme “Gauss-Laplace” vient du fait qu'elle est très semblable à l'approximation de Laplace [Barber 2012], mais approche la Hessienne de la même manière que le fait la méthode de Gauss-Newton vis à vis de la méthode de Newton.

Cette approximation de Gauss-Laplace intrinsèque est une généralisation de l'approximation de Gauss-Laplace euclidienne notamment utilisée dans [Bell 1993] pour obtenir les équations de l'IEKF. En effet, dans le cas où G est un espace euclidien, la méthode revient à ajuster une distribution normale multivariée.

Cette technique d'ajustement sera utilisée dans le reste de ce chapitre afin

d'obtenir les expressions du filtre et du lisseur proposés.

3.4 Algorithme de Gauss-Newton intrinsèque

L'approximation de Gauss-Laplace intrinsèque présentée dans le sous-chapitre précédent suppose que nous sommes capable de trouver le minimiseur \hat{X} du problème suivant :

$$\hat{X} = \underset{X \in G}{\operatorname{argmin}} \|\phi(X)\|_{\Sigma}^2 \quad (3.11)$$

où $\phi(\cdot) : G \rightarrow \mathbb{R}^m$ et $m > p$, où p est la dimension de G .

Lorsque le minimiseur d'un tel problème n'est pas trivial, il est courant de recourir à une technique d'optimisation itérative pour le trouver.

Nous présentons donc dans ce sous-chapitre une généralisation de l'algorithme de Gauss-Newton aux groupes de Lie qui permet de prendre en compte la nature des paramètres de manière intrinsèque.

Cet algorithme sera utilisé dans le reste du chapitre lorsque nous aurons recours à une approximation de Gauss-Laplace intrinsèque et que le minimiseur de (3.11) ne sera pas trivial.

3.4.1 Introduction à l'optimisation intrinsèque

Les techniques d'optimisation itératives permettant l'estimation de paramètres euclidiens, telle qu'une descente de gradient ou une méthode de Newton, consistent à mettre à jour itérativement les paramètres $x \in \mathbb{R}^n$ de la manière suivante :

$$x^{l+1} = x^l + \delta^{l+1/l} \quad (3.12)$$

où $\delta^{l+1/l} \in \mathbb{R}^n$ est un incrément qui corrige la précédente valeur des paramètres x^l afin d'obtenir leur nouvelle valeur x^{l+1} telle que $\rho(x^{l+1}) < \rho(x^l)$, où ρ est le critère à optimiser.

Cependant, une technique d'optimisation itérative sur un groupe de Lie G , prenant en compte la géométrie de G de manière intrinsèque, ne peut pas utiliser l'équation de mise à jour (3.12) puisque l'opérateur “+” ne garantit pas que les paramètres restent sur la variété.

L'ingrédient clé [Absil 2009] permettant de prendre en compte la géométrie du groupe de Lie est de remplacer l'équation (3.12) par :

$$X^{l+1} = \exp_G^{\wedge}(\delta^{l+1/l}) X^l \quad (3.13)$$

où $X^l, X^{l+1} \in G$ et $\delta^{l+1/l} \in \mathbb{R}^p$. L'utilisation de l'application exponentielle

de matrice permet en effet de s'assurer que la nouvelle valeur des paramètres demeure sur la variété.

Il est intéressant de noter la similarité entre la forme de cette équation de mise à jour (3.13) et l'expression d'une variable aléatoire suivant une distribution normale concentrée sur groupe de Lie (1.100).

3.4.2 Algorithme LG-GN

Un algorithme bien connu dans la littérature, permettant d'atteindre un minimum local du problème (3.11) lorsque G est un espace euclidien, est l'algorithme de Gauss-Newton. Nous proposons ici d'étendre ce formalisme aux groupes de Lie et appelons l'algorithme obtenu algorithme de Gauss-Newton intrinsèque sur groupe de Lie (LG-GN).

Remarquons qu'une généralisation de l'algorithme de Gauss-Newton dédiée aux variétés Riemanniennes a déjà été proposée dans [Absil 2009] p.126. Ainsi, l'algorithme que nous présentons peut s'inscrire dans ce formalisme et profiter de sa preuve de convergence. Cependant, nous proposons ici une version de l'algorithme dédiée aux groupes de Lie en utilisant uniquement sa structure de sous-groupe à un paramètre, c'est à dire sans exploiter *explicitement* la structure de variété Riemannienne du groupe de Lie.

Comme dans le cas euclidien, la convergence de l'algorithme dépend de la valeur initiale des paramètres X^0 . La valeur X^{l+1} des paramètres mis à jour est obtenue en appliquant un développement de Taylor sur groupe de Lie de la fonction $\phi(\cdot)$ autour de la précédente valeur X^l des paramètres. A chaque itération, il faut donc résoudre le problème suivant :

$$\delta^{l+1/l} = \operatorname{argmin}_{\delta \in \mathbb{R}^p} \left\| \phi(X^l) - J_l \delta \right\|_{\Sigma}^2 \quad (3.14)$$

où J_l est défini comme :

$$J_l = - \left. \frac{d\phi(\exp_G^\wedge(s) X^l)}{ds} \right|_{s=0} \quad (3.15)$$

Nous supposons ici que le rang de J_l est égal à p . Ainsi, la solution de ce problème est donnée par :

$$(J_l^T \Sigma^{-1} J_l) \delta^{l+1/l} = J_l^T \Sigma^{-1} \phi(X^l) \quad (3.16)$$

La valeur des paramètres est finalement mise à jour de la façon suivante :

$$X^{l+1} = \exp_G^\wedge(\delta^{l+1/l}) X^l \quad (3.17)$$

Afin de s'assurer de la convergence de l'algorithme, il est courant multiplier l'incrément $\delta^{l+1/l}$ par un facteur α^l lors de la mise à jour, où $0 < \alpha^l \leq 1$:

$$X^{l+1} = \exp_G^\wedge \left(\alpha^l \delta^{l+1/l} \right) X^l \quad (3.18)$$

3.5 Filtre de Kalman étendu itéré

Le filtre de Kalman étendu itéré est considéré comme une méthode de référence dans le domaine de l'estimation non linéaire de paramètres euclidiens à partir d'observations elles-mêmes euclidiennes. Dans ce sous-chapitre, nous proposons d'étendre cette approche au cas où l'état et les observations évoluent sur des groupes de Lie, en utilisant les outils présentés dans les sections précédentes, à savoir l'approximation de Gauss-Laplace intrinsèque et l'algorithme d'optimisation de Gauss-Newton intrinsèque.

3.5.1 Rappels sur le filtrage de Kalman étendu itéré sur espace euclidien

Le filtre de Kalman étendu itéré (IEKF) [Bell 1993 ; Bertsekas 1996] est un filtre dédié aux systèmes non linéaires. L'objectif de ce filtre est de récursivement approcher la distribution a posteriori $p(x_k | z_1, \dots, z_k)$ par une distribution normale multivariée. La variable $x_k \in \mathbb{R}^p$ est l'état que nous souhaitons estimer à l'instant k , reliée à x_{k-1} par le modèle de propagation :

$$x_k = f(x_{k-1}) + n_k \quad (3.19)$$

où $n_k \sim \mathcal{N}_{\mathbb{R}^p}(n_k; \mathbf{0}_{p \times 1}, R_k)$ est un bruit blanc gaussien et $f : \mathbb{R}^p \rightarrow \mathbb{R}^p$ est une fonction différentiable. La variable $z_k \in \mathbb{R}^q$ est l'observation disponible à l'instant k , reliée à x_k par le modèle d'observation :

$$z_k = h(x_k) + w_k \quad (3.20)$$

où $w_k \sim \mathcal{N}_{\mathbb{R}^q}(w_k; \mathbf{0}_{q \times 1}, Q_k)$ est un bruit blanc gaussien et $h(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}^q$ est une fonction différentiable. w_k et le bruit du modèle de propagation n_k sont supposés indépendants.

L'idée générale de ce filtre est d'approcher à chaque pas de temps la distribution a posteriori de l'état par une distribution normale multivariée en utilisant une approximation de Gauss-Laplace (voir paragraphe 3.3). Cette technique d'ajustement impose de trouver le mode de la distribution a posteriori qui est utilisé comme estimé de l'état. De cette manière, les étapes classiques du filtre de Kalman sont reformulées comme des problèmes d'optimisation. Le fait de

voir le filtrage de Kalman sous cet angle permet d'aboutir aux équations de l'IEKF qui produit en pratique de meilleurs résultats que le simple EKF.

Le filtre est composé de deux étapes.

3.5.1.1 Étape de propagation

L'étape de propagation consiste à approcher la distribution suivante :

$$\begin{aligned} & p(x_k | z_1, \dots, z_{k-1}) \\ &= \int p(x_k | x_{k-1}) p(x_{k-1} | z_1, \dots, z_{k-1}) d_L x_{k-1} \end{aligned} \quad (3.21)$$

$$\approx \mathcal{N}_{\mathbb{R}^p}(x_k; \mu_{k|k-1}, P_{k|k-1}) \quad (3.22)$$

La manière classique d'obtenir $\mu_{k|k-1}$ est de propager la moyenne précédente $\mu_{k-1|k-1}$ à travers le modèle de propagation (3.19) sans bruit. La covariance $P_{k|k-1}$ quant à elle est obtenue en propageant $P_{k-1|k-1}$ à travers le modèle de propagation linéarisé.

Une autre façon de voir l'étape de propagation est d'ajuster une distribution normale multivariée à l'intégrande dans (3.21) en utilisant une approximation de Gauss-Laplace (voir paragraphe 3.3) puis de marginaliser x_{k-1} . En effet, cherchons le minimum de la log-vraisemblance négative de l'intégrande de (3.21), où nous supposons que $p(x_{k-1} | z_1, \dots, z_{k-1}) = \mathcal{N}_{\mathbb{R}^p}(x_{k-1}; \mu_{k-1|k-1}, P_{k-1|k-1})$:

$$\{\hat{x}_k, \hat{x}_{k-1}\} = \underset{x_k \in \mathbb{R}^p, x_{k-1} \in \mathbb{R}^p}{\operatorname{argmin}} \|x_k - f(x_{k-1})\|_{R_k}^2 + \|x_{k-1} - \mu_{k-1|k-1}\|_{P_{k-1|k-1}}^2 \quad (3.23)$$

Une solution triviale de ce problème est $\hat{x}_{k-1} = \mu_{k-1|k-1}$ et $\hat{x}_k = f(\mu_{k-1|k-1})$. Il suffit alors de poser $\mu_{k|k-1} = \hat{x}_k$, ce qui est motivé par le fait que la moyenne coïncide avec le mode dans le cas gaussien. Finalement, $P_{k|k-1}$ est approchée en utilisant une approximation de Gauss-Laplace de (3.23) autour de son mode, ce qui conduit, dans le cas de (3.23), à une formule explicite pour la propagation de la covariance.

3.5.1.2 Étape de mise à jour

L'étape de mise à jour consiste à approcher la distribution suivante :

$$\begin{aligned} & p(x_k | z_1, \dots, z_k) \\ & \propto p(z_k | x_k) p(x_k | z_1, \dots, z_{k-1}) \end{aligned} \quad (3.24)$$

$$\approx \mathcal{N}_{\mathbb{R}^p}(x_k; \mu_{k|k}, P_{k|k}) \quad (3.25)$$

A nouveau, les équations de mise à jour de l'IEKF peuvent être obtenues en ajustant une distribution gaussienne à (3.24) à l'aide d'une approximation de Gauss-Laplace (voir 3.3).

En effet, définissons $\mu_{k|k} = \hat{x}_k$ comme le minimiseur de la log-vraisemblance négative de (3.24) :

$$\hat{x}_k = \operatorname{argmin}_{x_k \in \mathbb{R}^p} \left(\|z_k - h(x_k)\|_{Q_k}^2 + \|x_k - \mu_{k|k-1}\|_{P_{k|k-1}}^2 \right) \quad (3.26)$$

Une manière de minimiser (3.26) est d'utiliser l'algorithme de Gauss-Newton, ce qui nous permet de calculer $\mu_{k|k}$, alors que $P_{k|k}$ peut être obtenu en effectuant une approximation de Laplace de (3.26) autour de $\mu_{k|k}$. Or, en tirant parti de la structure spécifique du GN appliqué à (3.26), il est possible d'aboutir aux expressions de mise à jour de l'IEKF [Bell 1993].

3.5.1.3 Algorithme IEKF

L'algorithme IEKF est présenté ci-après :

Algorithme 3.1 IEKF

Entrée : $\mu_{k-1|k-1}$, $P_{k-1|k-1}$, z_k

Sortie : $\mu_{k|k}$, $P_{k|k}$

Propagation :

$$\mu_{k|k-1} = f(\mu_{k-1|k-1})$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + R_k \text{ où } F_k = \left. \frac{d}{dx} f(x) \right|_{x=\mu_{k-1|k-1}}$$

Mise à jour :

$$\text{Poser } x^0 = \mu_{k|k-1}$$

Itérer jusqu'à convergence

$$K_l = P_{k|k-1} H_l^T (H_l P_{k|k-1} H_l^T + Q_k)^{-1} \text{ où } H_l = \left. \frac{d}{dx} h(x) \right|_{x=x^l}$$

$$\tilde{z}_l = z_k - h(x^l)$$

$$m_l^- = K_l (\tilde{z}_l + H_l (x^l - \mu_{k|k-1}))$$

$$x^{l+1} = \mu_{k|k-1} + m_l^-$$

A la convergence

$$\mu_{k|k} = x^l$$

$$P_{k|k} = (Id - K_l H_l) P_{k|k-1}$$

Cet algorithme, qui est considéré comme une méthode de référence dans le domaine de l'estimation non linéaire, est restreint au cas où l'état et les observations évoluent sur des espaces euclidiens. Dans la suite de cette partie, nous proposons d'étendre cette approche au cas où l'état et les observations évoluent sur des groupes de Lie.

3.5.2 Modèles de propagation et d'observation proposés

Afin de généraliser l'IEKF aux groupes de Lie, il convient de généraliser les modèles de propagation et de mise à jour afin de prendre en compte le fait que l'état et les observations évoluent sur des groupes de Lie.

3.5.2.1 Modèle de propagation

Nous considérons l'équation de propagation suivante qui est une généralisation directe du modèle de propagation (3.19) aux groupes de Lie :

$$X_k = \exp_G^\wedge(n_k) f(X_{k-1}) \quad (3.27)$$

où $X_k \in G$ est l'état que nous souhaitons estimer et G est un groupe de Lie matriciel de dimension p . La variable $n_k \sim \mathcal{N}_{\mathbb{R}^p}(n_k; \mathbf{0}_{p \times 1}, R_k)$ est un bruit blanc gaussien. La fonction $f : G \rightarrow G$ est supposée différentiable.

Remarquons tout d'abord que ce modèle induit la distribution conditionnelle suivante :

$$p(X_k | X_{k-1}) = \mathcal{N}_G^R(X_k; f(X_{k-1}), R_k) \quad (3.28)$$

Injecter le bruit de l'autre côté dans (3.27) aurait induit une distribution conditionnelle ayant la forme d'une distribution normale concentrée à gauche. Nous avons ici choisi de travailler avec des distributions "à droite", mais nous aurions pu faire mener les mêmes développements avec des distributions "à gauche", comme dans le chapitre précédent, et les expressions obtenues auraient été très similaires.

Remarquons également que le modèle que nous considérons ici est plus générique que celui considéré pour le D-LG-EKF car la fonction f permet de représenter une plus grande diversité de modèles.

3.5.2.2 Modèle d'observation

Le modèle d'observation que nous considérons est le "même" que celui du CD-LG-EKF (voir paragraphe 2.3.2.2) à cela près que le bruit est injecté à gauche. Plus précisément, nous considérons des observations discrètes vivant sur un groupe de Lie matriciel G' de dimension q de la forme :

$$Z_k = \exp_{G'}^\wedge(w_k) h(X_k) \quad (3.29)$$

où $h : G \rightarrow G'$ est une fonction différentiable et $w_k \sim \mathcal{N}_{\mathbb{R}^q}(w_k; \mathbf{0}_{q \times 1}, Q_k)$ est un bruit blanc gaussien.

Remarquons que ce modèle induit la distribution conditionnelle suivante :

$$p(Z_k | X_k) = \mathcal{N}_{G'}^R(Z_k; h(X_k), Q_k) \quad (3.30)$$

3.5.3 Solution proposée

Afin d'estimer l'état X_k , nous proposons de supposer que sa distribution a posteriori a la forme d'une distribution normale concentrée (à droite) :

$$p(X_k|Z_1, \dots, Z_l) \approx \mathcal{N}_G^R(X_k; \mu_{k|l}, P_{k|l}) \quad (3.31)$$

Plus précisément, nous nous intéressons aux cas où $l = k - 1$ (étape de propagation) et $l = k$ (étape de mise à jour). Ainsi, notre objectif est de montrer comment propager et mettre à jour les paramètres $\mu_{k|k}$ et $P_{k|k}$ de la distribution normale concentrée. Dans notre formalisme, nous choisissons comme estimateur de l'état X la “moyenne” de la distribution normale concentrée.

3.5.4 Propagation

Commençons par supposer que :

$$p(X_{k-1}|Z_1, \dots, Z_{k-1}) = \mathcal{N}_G^R(X_{k-1}; \mu_{k-1|k-1}, P_{k-1|k-1}) \quad (3.32)$$

L'objectif de cette section est de montrer comment ajuster une distribution normale concentrée à la distribution a posteriori de l'état propagé à l'aide d'une approximation de Gauss-Laplace (voir paragraphe 3.3), c'est à dire :

$$p(X_k|Z_1, \dots, Z_{k-1}) = \int p(X_k|X_{k-1}) p(X_{k-1}|Z_1, \dots, Z_{k-1}) d_H X_{k-1} \quad (3.33)$$

$$\approx \mathcal{N}_G^R(X_k; \mu_{k|k-1}, P_{k|k-1}) \quad (3.34)$$

Nous proposons ici d'ajuster une distribution normale concentrée à l'intégrande de (3.33) puis de marginaliser X_{k-1} .

3.5.4.1 Propagation de la moyenne

Afin de propager la moyenne, nous minimisons la log-vraisemblance négative de l'intégrande de (3.33) :

$$\begin{aligned} \{\hat{X}_k, \hat{X}_{k-1}\} &= \operatorname{argmin}_{X_k \in G, X_{k-1} \in G} -\log(p(X_k|X_{k-1}) p(X_{k-1}|Z_1, \dots, Z_{k-1})) \\ &= \operatorname{argmin}_{X_k \in G, X_{k-1} \in G} \left(\left\| \log_G^\vee(X_k f(X_{k-1})^{-1}) \right\|_{R_k}^2 + \left\| \log_G^\vee(X_{k-1} \mu_{k-1|k-1}^{-1}) \right\|_{P_{k-1|k-1}}^2 \right) \end{aligned} \quad (3.35)$$

Une solution triviale à ce problème est $\hat{X}_{k-1} = \mu_{k-1|k-1}$ et $\hat{X}_k = f(\mu_{k-1|k-1})$. Et nous prenons finalement comme estimé pour la moyenne :

$$\mu_{k|k-1} = \hat{X}_k = f(\mu_{k-1|k-1}) \quad (3.36)$$

3.5.4.2 Propagation de la covariance

Concernant la propagation de la covariance, nous procédons à l'approximation de Gauss-Laplace. Nous linéarisons la fonction d'erreur présente à l'intérieur de la norme de Mahalanobis dans (3.35) autour de son minimiseur. Nous avons alors :

$$-\log(p(X_k|X_{k-1})p(X_{k-1}|Z_1, \dots, Z_{k-1})) \approx \left\| \begin{bmatrix} Id & -F_k \\ \mathbf{0} & Id \end{bmatrix} \delta \right\|_{\Sigma}^2 \quad (3.37)$$

où $\delta = \begin{bmatrix} \delta_k^T & \delta_{k-1}^T \end{bmatrix}^T$, $X_{k-1} = \exp_G^\wedge(\delta_{k-1}) \mu_{k-1|k-1}$, $X_k = \exp_G^\wedge(\delta_k) f(\mu_{k-1|k-1})$,

$$\Sigma = \begin{bmatrix} R_k & \\ & P_{k-1|k-1} \end{bmatrix} \quad (3.38)$$

et

$$F_k = - \frac{d \log_G^\vee \left(f(\mu_{k-1|k-1}) f(\exp_G^\wedge(s) \mu_{k-1|k-1})^{-1} \right)}{ds} \Big|_{s=\mathbf{0}} \quad (3.39)$$

L'équation (3.37) a alors la forme suivante :

$$\delta^T \begin{bmatrix} R_k^{-1} & -R_k^{-1} F_k \\ -F_k^T R_k^{-1} & F_k^T R_k^{-1} F_k + P_{k-1|k-1}^{-1} \end{bmatrix} \delta \quad (3.40)$$

et nous choisissons comme valeur pour la covariance P :

$$P = \begin{bmatrix} R_k^{-1} & -R_k^{-1} F_k \\ -F_k^T R_k^{-1} & F_k^T R_k^{-1} F_k + P_{k-1|k-1}^{-1} \end{bmatrix}^{-1} \quad (3.41)$$

$$= \begin{bmatrix} F_k P_{k-1|k-1} F_k^T + R_k & F_k P_{k-1|k-1} \\ P_{k-1|k-1} F_k^T & P_{k-1|k-1} \end{bmatrix} \quad (3.42)$$

où nous avons utilisé le lemme d'inversion matricielle (F.1). Sous l'hypothèse de distribution normale concentrée, le bloc supérieur gauche de P dans (3.42) correspond à la covariance propagée $P_{k|k-1}$ (voir annexe A). Ainsi, nous avons obtenu l'équation de propagation de la covariance :

$$P_{k|k-1} = R_k + F_k P_{k-1|k-1} F_k^T \quad (3.43)$$

3.5.4.3 Résumé de l'étape de propagation

L'étape de propagation consiste, comme son nom l'indique, à propager la moyenne $\mu_{k-1|k-1}$ et la covariance $P_{k-1|k-1}$ en utilisant le modèle de propagation (3.27) afin d'obtenir la moyenne $\mu_{k|k-1}$ et la covariance $P_{k|k-1}$. A la fin de l'étape de propagation, la distribution a posteriori approchée de l'état est alors paramétrée de la manière suivante :

$$p(X_k|Z_1, \dots, Z_{k-1}) \approx \mathcal{N}_G^R(X_k; \mu_{k|k-1}, P_{k|k-1}) \quad (3.44)$$

3.5.5 Mise à jour

A l'issue de l'étape de propagation nous avons :

$$p(X_k|Z_1, \dots, Z_{k-1}) \approx \mathcal{N}_G^R(X_k; \mu_{k|k-1}, P_{k|k-1}) \quad (3.45)$$

L'objectif de cette partie est de montrer comment ajuster une distribution normale concentrée à la distribution a posteriori de l'état après réception de l'observation Z_k à l'aide d'une approximation de Gauss-Laplace (voir paragraphe 3.3), c'est à dire :

$$p(X_k|Z_1, \dots, Z_k) \propto p(Z_k|X_k) p(X_k|Z_1, \dots, Z_{k-1}) \quad (3.46)$$

$$\approx \mathcal{N}_G^R(X_k; \mu_{k|k}, P_{k|k}) \quad (3.47)$$

3.5.5.1 Mise à jour de la moyenne

Afin de mettre à jour la moyenne, nous choisissons de minimiser la log-vraisemblance négative de (3.46) :

$$\hat{X}_k = \underset{X_k \in G}{\operatorname{argmin}} \left(\left\| \log_{G'}^\vee(Z_k h(X_k)^{-1}) \right\|_{Q_k}^2 + \left\| \log_G^\vee(X_k \mu_{k|k-1}^{-1}) \right\|_{P_{k|k-1}}^2 \right) \quad (3.48)$$

Afin de minimiser cette fonction, nous proposons d'employer l'algorithme LG-GN. Pour cela, nous introduisons les notations suivantes :

$$\delta^{l+1/l} = \log_G^\vee(X^{l+1} (X^l)^{-1}) \quad (3.49)$$

$$\delta^l = \log_G^\vee(X^l \mu_{k|k-1}^{-1}) \quad (3.50)$$

et

$$\delta^{l+1} = \log_G^\vee(X^{l+1} \mu_{k|k-1}^{-1}) \quad (3.51)$$

où X^l représente la valeur des paramètres à l'itération l du LG-GN.

A l'itération l , nous cherchons la solution du problème suivant :

$$\begin{aligned}\delta^{l+1/l} &= \underset{\delta \in \mathbb{R}^p}{\operatorname{argmin}} \left(\left\| \log_{G'}^\vee \left(Z_k h \left(X^l \right)^{-1} \right) - H_l \delta \right\|_{Q_k}^2 \right. \\ &\quad \left. + \left\| \delta^l + \varphi_G \left(\delta^l \right) \delta \right\|_{P_{k|k-1}}^2 \right) \\ &= \underset{\delta \in \mathbb{R}^p}{\operatorname{argmin}} \left\| \psi \left(X^l \right) - \Psi_l \delta \right\|_{\Xi_k}^2\end{aligned}\quad (3.52)$$

où $\varphi_G \left(\delta^l \right)$ est défini dans l'équation (1.33),

$$H_l = - \left. \frac{d \log_{G'}^\vee \left(Z_k h \left(\exp_G^\wedge(s) X^l \right)^{-1} \right)}{ds} \right|_{s=0} \quad (3.53)$$

$$\psi \left(X^l \right) = \left[\log_{G'}^\vee \left(Z_k h \left(X^l \right)^{-1} \right)^T \quad \left(\delta^l \right)^T \right]^T \quad (3.54)$$

$$\Psi_l = \left[H_l^T \quad -\varphi_G \left(\delta^l \right)^T \right]^T \quad (3.55)$$

$$\Xi_k = \begin{bmatrix} Q_k & \mathbf{0} \\ \mathbf{0} & P_{k|k-1} \end{bmatrix} \quad (3.56)$$

Pour le reste de cette partie, nous adoptons la notation suivante concernant la jacobienne à gauche de G et son inverse : $\varphi_G \left(\delta^l \right) \equiv \varphi_l$ et $\Phi_l = \varphi_l^{-1}$.

La solution de (3.52) est donnée par :

$$\begin{aligned}\delta^{l+1/l} &= \left(\Psi_l^T \Xi_k^{-1} \Psi_l \right)^{-1} \Psi_l^T \Xi_k^{-1} \psi \left(X^l \right) \\ &= \left(H_l^T Q_k^{-1} H_l + \varphi_l^T P_{k|k-1}^{-1} \varphi_l \right)^{-1} \left\{ H_l^T Q_k^{-1} \log_{G'}^\vee \left(Z_k h \left(X^l \right)^{-1} \right) - \varphi_l^T P_{k|k-1}^{-1} \delta^l \right\}\end{aligned}\quad (3.57)$$

Nous allons maintenant montrer qu'en négligeant les termes d'ordre 2 en $\delta^{l+1/l}$ cette solution peut être réécrite pour donner lieu à une généralisation de l'équation de mise à jour de la moyenne de l'IEKF aux groupes de Lie. En

effet :

$$\begin{aligned}
\delta^{l+1} &= \log_G^\vee \left(X^{l+1} (X^l)^{-1} X^l \mu_{k|k-1}^{-1} \right) \\
&= \log_G^\vee \left(\exp_G^\wedge (\delta^{l+1/l}) \exp_G^\wedge (\delta^l) \right) \\
&\simeq \varphi_l \delta^{l+1/l} + \delta^l \\
&= \varphi_l \left(H_l^T Q_k^{-1} H_l + \varphi_l^T P_{k|k-1}^{-1} \varphi_l \right)^{-1} \left\{ H_l^T Q_k^{-1} \log_{G'}^\vee \left(Z_k h (X^l)^{-1} \right) - \varphi_l^T P_{k|k-1}^{-1} \delta^l \right\} + \delta^l \\
&= \varphi_l \left(H_l^T Q_k^{-1} H_l + \varphi_l^T P_{k|k-1}^{-1} \varphi_l \right)^{-1} \\
&\quad \left\{ H_l^T Q_k^{-1} \log_{G'}^\vee \left(Z_k h (X^l)^{-1} \right) - \varphi_l^T P_{k|k-1}^{-1} \delta^l + \left(H_l^T Q_k^{-1} H_l + \varphi_l^T P_{k|k-1}^{-1} \varphi_l \right) \Phi_l \delta^l \right\} \\
&= \varphi_l \left(H_l^T Q_k^{-1} H_l + \varphi_l^T P_{k|k-1}^{-1} \varphi_l \right)^{-1} H_l^T Q_k^{-1} \left\{ \log_{G'}^\vee \left(Z_k h (X^l)^{-1} \right) + H_l \Phi_l \delta^l \right\} \\
&= K_l \left\{ \log_{G'}^\vee \left(Z_k h (X^l)^{-1} \right) + H_l \Phi_l \delta^l \right\} \tag{3.58}
\end{aligned}$$

où K_l est appelé gain de Lie-Kalman et s'exprime de la manière suivante (nous omettons les indices pour cette démonstration) :

$$\begin{aligned}
K &= \varphi \left(H^T Q^{-1} H + \varphi^T P^{-1} \varphi \right)^{-1} H^T Q^{-1} \\
&= \varphi \left(H^T Q^{-1} H + \varphi^T P^{-1} \varphi \right)^{-1} \left(H^T Q^{-1} (H \Phi P \Phi^T H^T + Q) (H \Phi P \Phi^T H^T + Q)^{-1} \right) \\
&= \varphi \left(H^T Q^{-1} H + \varphi^T P^{-1} \varphi \right)^{-1} \left((H^T Q^{-1} H \Phi P \Phi^T H^T + H^T) (H \Phi P \Phi^T H^T + Q)^{-1} \right) \\
&= \varphi \left(H^T Q^{-1} H + \varphi^T P^{-1} \varphi \right)^{-1} \left(((H^T Q^{-1} H + \varphi^T P^{-1} \varphi) \Phi P \Phi^T H^T) (H \Phi P \Phi^T H^T + Q)^{-1} \right) \\
&= P \Phi^T H^T (H \Phi P \Phi^T H^T + Q)^{-1} \tag{3.59}
\end{aligned}$$

Cependant (3.58) peut être simplifiée en remarquant que :

$$\Phi_l \delta^l = \Phi_G (\delta^l) \delta^l = \sum_{n=0}^{\infty} \frac{1}{(n+1)!} \text{ad}_G (\delta^l)^n \delta^l = \delta^l \tag{3.60}$$

car

$$\text{ad}_G (\delta^l) \delta^l = \mathbf{0} \tag{3.61}$$

Ainsi, l'équation de mise à jour des paramètres du LG-GN peut s'écrire de la forme suivante :

$$X^{l+1} = \exp_G^\wedge (\delta^{l+1/l}) X^l \approx \exp_G^\wedge \left(K_l \left\{ \log_{G'}^\vee \left(Z_k h (X^l)^{-1} \right) + H_l \delta^l \right\} \right) \mu_{k|k-1} \tag{3.62}$$

A la convergence de l'algorithme, nous choisissons $\mu_{k|k} = \hat{X}_k$.

3.5.5.2 Mise à jour de la covariance

Afin d'estimer la covariance mise à jour, nous proposons d'utiliser une approximation de Gauss-Laplace intrinsèque. Pour cela, nous linéarisons la fonction d'erreur présente à l'intérieur de la norme de Mahalanobis dans (3.48) autour de son minimiseur :

$$-\log(p(Z_k|X_k)p(X_k|Z_1, \dots, Z_{k-1})) \approx \left\| \begin{bmatrix} \log_{G'}^\vee \left(Z_k h(\mu_{k|k})^{-1} \right) - H_l \delta \\ \delta^l + \varphi_l \delta \end{bmatrix} \right\|_\Sigma^2 \quad (3.63)$$

où $\Sigma = \begin{bmatrix} Q_k & \mathbf{0} \\ \mathbf{0} & P_{k|k-1} \end{bmatrix}$ et $X_k = \exp_G^\wedge(\delta) \mu_{k|k}$. Dans l'équation précédente, le terme quadratique a la forme suivante :

$$\delta^T (H_l^T Q_k^{-1} H_l + \varphi_l^T P_{k|k-1}^{-1} \varphi_l) \delta \quad (3.64)$$

et nous choisissons alors comme valeur pour la covariance $P_{k|k}$:

$$P_{k|k} = (H_l^T Q_k^{-1} H_l + \varphi_l^T P_{k|k-1}^{-1} \varphi_l)^{-1} \quad (3.65)$$

Cependant il est possible de montrer que (nous omettons les indices pour cette démonstration) :

$$\begin{aligned} P_{k|k} &= (H^T Q^{-1} H + \varphi^T P^{-1} \varphi)^{-1} \\ &= (H^T Q^{-1} H + \varphi^T P^{-1} \varphi)^{-1} \{ (H^T Q^{-1} H + \varphi^T P^{-1} \varphi) \Phi P \Phi^T - H^T Q^{-1} H \Phi P \Phi^T \} \\ &= \Phi P \Phi^T - (H^T Q^{-1} H + \varphi^T P^{-1} \varphi)^{-1} H^T Q^{-1} H \Phi P \Phi^T \\ &= \Phi P \Phi^T - \Phi K H \Phi P \Phi^T \\ &= \Phi (Id - K H \Phi) P \Phi^T \end{aligned} \quad (3.66)$$

Ainsi, nous avons obtenu l'équation de mise à jour de la covariance qui donne lieu à une généralisation de l'équation de mise à jour de la covariance de l'IEKF aux groupes de Lie :

$$P_{k|k} = \Phi_l (Id - K_l H_l \Phi_l) P_{k|k-1} \Phi_l^T \quad (3.67)$$

3.5.5.3 Résumé de l'étape de mise à jour

L'étape de mise à jour consiste, comme son nom l'indique, à mettre à jour la moyenne $\mu_{k|k-1}$ et la covariance $P_{k|k-1}$ en incorporant l'information contenue dans l'observation Z_k qui est reliée à l'état par le modèle d'observation (3.29) afin d'obtenir la moyenne $\mu_{k|k}$ et la covariance $P_{k|k}$. A la fin de l'étape de mise

à jour, la distribution a posteriori approchée de l'état est alors paramétrée de la manière suivante :

$$p(X_k | Z_1, \dots, Z_k) \approx \mathcal{N}_G^R(X_k; \mu_{k|k}, P_{k|k}) \quad (3.68)$$

3.5.6 Algorithme LG-IEKF

L'algorithme formé par les étapes de propagation et de mise à jour présentées précédemment est appelé filtre de Kalman étendu itéré sur groupe de Lie (LG-IEKF).

3.5.6.1 Vue générale

L'algorithme LG-IEKF est résumé ci-après (voir paragraphe 3.5.4 et paragraphe 3.5.5 pour plus de détails) :

Algorithme 3.2 LG-IEKF

Entrée : $\mu_{k-1|k-1}, P_{k-1|k-1}, Z_k$

Sortie : $\mu_{k|k}, P_{k|k}$

Propagation :

$$\mu_{k|k-1} = f(\mu_{k-1|k-1})$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + R_k$$

Mise à jour :

$$\text{Poser } X^0 = \mu_{k|k-1}$$

Itérer jusqu'à convergence

$$K_l = P_{k|k-1} \Phi_l^T H_l^T (H_l \Phi_l P_{k|k-1} \Phi_l^T H_l^T + Q_k)^{-1}$$

$$\tilde{z}_l = \log_{G'}^\vee \left(Z_k h(X^l)^{-1} \right)$$

$$m_l^- = K_l \left\{ \tilde{z}_l + H_l \log_{G'}^\vee \left(X^l \mu_{k|k-1}^{-1} \right) \right\}$$

$$X^{l+1} = \exp_G^\wedge(m_l^-) \mu_{k|k-1}$$

A la convergence

$$\mu_{k|k} = X^l$$

$$P_{k|k} = \Phi_l (Id - K_l H_l \Phi_l) P_{k|k-1} \Phi_l^T$$

3.5.6.2 Une généralisation de l'IEKF

Dans le cas spécifique où G , le groupe de Lie sur lequel évolue l'état X , et G' , le groupe de Lie sur lequel évoluent les observations, sont des espaces euclidiens, alors l'algorithme LG-IEKF se réduit à l'IEKF. Cette propriété vient du fait que le modèle de propagation devient une fonction non linéaire de l'état

X_{k-1} avec un bruit additif, et l'équation d'observation prend la forme d'une fonction non linéaire de l'état X_k avec un bruit additif.

L'étape de propagation d l'IEKF est identique à celle du LG-IEKF à cela près que la matrice F_k du LG-IEKF est une dérivée utilisant la notion de sous-groupe à un paramètre qui se réduit à une jacobienne classique si G est un espace euclidien. Concernant la mise à jour, le LG-IEKF et l'IEKF sont très proches. Nous observons cependant des matrices Φ_G qui effectuent des reparamétrisations de la covariance (voir paragraphe 1.2.2), en effet :

$$\begin{aligned} P_{k|k} &= \Phi_l (Id - K_l H_l \Phi_l) P_{k|k-1} \Phi_l^T \\ &= \Phi_l P_{k|k-1} \Phi_l^T - \Phi_l P_{k|k-1} \Phi_l^T H_l^T \left(H_l \Phi_l P_{k|k-1} \Phi_l^T H_l^T + Q_k \right)^{-1} H_l \Phi_l P_{k|k-1} \Phi_l^T \\ &= P_{k|k-1}^{rep} - P_{k|k-1}^{rep} H_l^T \left(H_l P_{k|k-1}^{rep} H_l^T + Q_k \right)^{-1} H_l P_{k|k-1}^{rep} \end{aligned} \quad (3.69)$$

où

$$\begin{aligned} P_{k|k-1}^{rep} &= \Phi_l P_{k|k-1} \Phi_l^T \\ &= \Phi_G \left(\log_G^\vee \left(\mu_{k|k} \mu_{k|k-1}^{-1} \right) \right) P_{k|k-1} \Phi_G \left(\log_G^\vee \left(\mu_{k|k} \mu_{k|k-1}^{-1} \right) \right)^T \end{aligned} \quad (3.70)$$

correspond à la matrice de covariance issue de l'étape de propagation reparamétrisée par Φ_l afin qu'elle ait un *sens* (voir paragraphe 1.2.2.2) non plus vis à vis de $\mu_{k|k-1}$ mais de $\mu_{k|k}$.

Remarquons également qu'au lieu d'un incrément additif (“+”), le LG-IEKF utilise un incrément faisant intervenir “ \exp^\wedge ”. De la même manière, au lieu de considérer une erreur euclidienne (“−”), le LG-IEKF emploie l'application “ \log^\vee ”. Concernant H_l , la même remarque que celle que nous avons faites concernant F_k peut être effectuée.

En ce sens, le LG-IEKF peut donc être vu comme une généralisation de l'IEKF.

3.5.6.3 Lien avec le D-LG-EKF

La mise à profit de l'approximation de Gauss-Laplace et du LG-GN nous a permis de proposer l'étape de mise à jour du LG-IEKF qui raffine le point de linéarisation au cours des itérations contrairement à l'étape de mise à jour du D-LG-EKF qui n'effectue qu'une seule linéarisation.

En outre, le modèle de propagation que nous avons considéré pour le LG-IEKF est plus générique que celui considéré pour le D-LG-EKF car la fonction f permet de modéliser une plus grande diversité de modèles.

En ce sens, le LG-IEKF peut être vu comme une extension du D-LG-EKF.

Cependant, afin d'étudier rigoureusement l'intérêt des itérations lors de la

mise à jour, nous introduisons un troisième algorithme que nous appelons LG-EKF qui correspond exactement au LG-IEKF du point de vue de la propagation mais qui n'effectue qu'une seule itération lors de la mise à jour.

3.5.6.4 Fenêtrage statistique

L'algorithme LG-IEKF que nous avons proposé repose sur un modèle d'observation où le bruit injecté est supposé blanc et gaussien. Cette hypothèse nous a permis de développer un algorithme à la fois élégant et efficace. Cependant, cette hypothèse rend le LG-IEKF peu robuste aux observations aberrantes, ce qui est également le cas pour les filtres de Kalman sur espace euclidien. Afin de rendre l'algorithme plus robuste, nous proposons d'effectuer un fenêtrage statistique [Ramachandr 2000] avant l'étape de mise à jour, qui consiste à tester, vis-à-vis de la distribution a posteriori courante de l'état, si l'observation reçue est aberrante ou non. Si la donnée est classée comme aberrante, alors l'étape de mise à jour n'est pas appliquée. Nous détaillons maintenant ce fenêtrage statistique.

Une observation Z_k n'est pas une donnée aberrante si :

$$\left\| \log_{G'}^\vee \left(Z_k h(X_k)^{-1} \right) \right\|_{Q_k}^2 < t \quad (3.71)$$

où t est un seuil que nous allons définir. Cependant, la vraie valeur de l'état X_k est inconnue puisque l'objectif du LG-IEKF est de l'estimer. Nous possédons seulement une approximation de sa distribution a posteriori :

$$p(X_k | Z_1, \dots, Z_{k-1}) \approx \mathcal{N}_G^R(X_k; \mu_{k|k-1}, P_{k|k-1}) \quad (3.72)$$

C'est à dire :

$$X_k = \exp_G^\wedge(\epsilon_{k|k-1}) \mu_{k|k-1} \quad (3.73)$$

où $\epsilon_{k|k-1} \sim \mathcal{N}_{\mathbb{R}^p}(\epsilon; \mathbf{0}_{p \times 1}, P_{k|k-1})$. Ainsi, afin de décider si Z_k est aberrant ou non, nous proposons le test suivant :

$$\left\| \log_{G'}^\vee \left(Z_k h(\mu_{k|k-1})^{-1} \right) \right\|_{Q_{err}}^2 < t \quad (3.74)$$

où

$$\begin{aligned} Q_{err} &= \mathbb{E} \left(\left(H_{k|k-1} \epsilon_{k|k-1} + w_k \right) \left(H_{k|k-1} \epsilon_{k|k-1} + w_k \right)^T \right) \\ &= H_{k|k-1} P_{k|k-1} H_{k|k-1}^T + Q_k \end{aligned} \quad (3.75)$$

En effet, en partant du modèle d'observation (3.29) et en négligeant les termes

d'ordre 2 en w_k et $\epsilon_{k|k-1}$, nous avons :

$$\begin{aligned}
 \mathbf{0} &= \log_{G'}^\vee \left(Z_k h(X_k)^{-1} \exp_{G'}^\wedge(-w_k) \right) \\
 &\simeq \log_{G'}^\vee \left(Z_k h \left(\exp_G^\wedge(\epsilon_{k|k-1}) \mu_{k|k-1} \right)^{-1} \exp_{G'}^\wedge(-w_k) \right) \\
 &\simeq \log_{G'}^\vee \left(Z_k h \left(\exp_G^\wedge(\epsilon_{k|k-1}) \mu_{k|k-1} \right)^{-1} \right) - w_k \\
 &\simeq \log_{G'}^\vee \left(Z_k h \left(\mu_{k|k-1} \right)^{-1} \right) - H_{k|k-1} \epsilon_{k|k-1} - w_k
 \end{aligned} \tag{3.76}$$

où

$$H_{k|k-1} = - \left. \frac{d \log_{G'}^\vee \left(Z_k h \left(\exp_G^\wedge(s) \mu_{k|k-1} \right)^{-1} \right)}{ds} \right|_{s=\mathbf{0}} \tag{3.77}$$

Ainsi, sous l'hypothèse de distribution gaussienne concentrée, le LHS (3.74) suit la loi du \mathcal{X}^2 avec q degrés de liberté. Par conséquent, une manière de décider si Z_k est une donnée aberrante ou non est de définir le seuil t comme une valeur p du $\chi^2(q)$ [Fisher 1949].

Remarquons cependant que puisque nous avons négligé les termes d'ordre 2, la valeur p théorique peut potentiellement être trop “stricte” en pratique.

3.5.7 Application du LG-IEKF au débruitage de transformations relatives

Nous avons choisi d'appliquer le LG-IEKF au problème du débruitage de transformations relatives qui consiste à estimer les transformations absolues à partir de transformations relatives bruitées.

Un tel problème apparaît par exemple dans le cadre de l'alignement de poses de caméra [Agrawal 2006]. Dans ce cas précis, une transformation est une matrice du groupe de Lie des mouvements d'un solide rigide ($SE(3)$). Les observations sont donc des transformations rigides entre deux poses de caméra et les transformations absolues que nous souhaitons estimer sont les transformations rigides entre la pose d'une caméra de référence et toutes les autres poses de caméra.

Ici nous abordons le problème générique où une transformation est définie comme un élément d'un groupe de Lie. Par conséquent, la solution que nous proposons s'applique non seulement au problème précédemment exposé mais également aux problèmes de la synchronisation de rotations ($SO(3)$) [Boumal 2013b], de la construction d'une mosaïque d'images ($SL(3)$) [Meidow 2011] ou encore de l'alignement de reconstructions 3D partielles ($Sim(3)$) [Wachinger 2008].

3.5.7.1 Problème et objectif

Nous souhaitons estimer les transformations absolues $\{X_{iR}\}_{i=1:N}$, où chaque transformation absolue $X_{iR} \in G'$ est définie comme la transformation entre un référentiel global R et un référentiel i , G' étant un groupe de Lie de dimension q .

Nous considérons le cas où les bruits sur les transformations relatives observées $\{Z_{ij}\}_{1 \leq i < j \leq N}$ sont mutuellement indépendants. Chaque $Z_{ij} \in G'$ représente une transformation relative bruitée, entre un référentiel j et un référentiel i , reliée aux transformations absolues par le modèle suivant :

$$Z_{ij} = \exp_{G'}^{\wedge} \left(b_{ij}^i \right) X_{iR} X_{jR}^{-1} \quad (3.78)$$

où $b_{ij}^i \sim \mathcal{N}_{\mathbb{R}^q} \left(b_{ij}^i; \mathbf{0}_{q \times 1}, \Sigma_{ij}^i \right)$ est un bruit blanc gaussien. Le problème considéré est illustré graphiquement en figure 3.1a.

Remarquons que l'équation (3.78) est invariante à l'action à droite de G' . C'est à dire, $X_{iR} Y (X_{jR} Y)^{-1} = X_{iR} X_{jR}^{-1}$ pour tout $Y \in G'$. Dans le contexte de l'alignement de poses de caméra par exemple, cela traduit la symétrie sous-jacente du problème, à savoir que le fait d'appliquer une rotation et une translation aux poses absolues des caméras n'affecte en rien les poses relatives observées.

3.5.7.2 Cas 1 : Estimation sans données aberrantes

Dans cette partie, nous traitons le cas “simple” où les observations ne contiennent pas de données aberrantes.

État de l'art Divers travaux traitent du problème auquel nous nous intéressons dans cette partie. Cependant, en général, ils ne prennent pas intrinsèquement en compte la géométrie sous-jacente du groupe de Lie sur lequel évoluent les paramètres et sont souvent adaptés à une application spécifique telle que :

- Débruitage d'orientations relatives (groupe de Lie $SO(3)$, ou $SU(2)$ quaternion unitaire), également connu sous le nom de moyenne de rotation multiples [Hartley 2013], ou encore synchronisation de rotations [Boumal 2013a]. Dans [Govindu 2001] et [Martinec 2007], afin d'obtenir un algorithme efficace requérant uniquement la résolution d'un grand système linéaire, les contraintes associées à la structure du groupe de Lie sont intentionnellement omises. [Fredriksson 2012] utilise une formulation similaire à celle de [Govindu 2001], mais les contraintes associées au groupe de Lie $SU(2)$ sont prises en compte de manière extrinsèque par le biais d'une approche basée sur la dualité Lagrangienne.

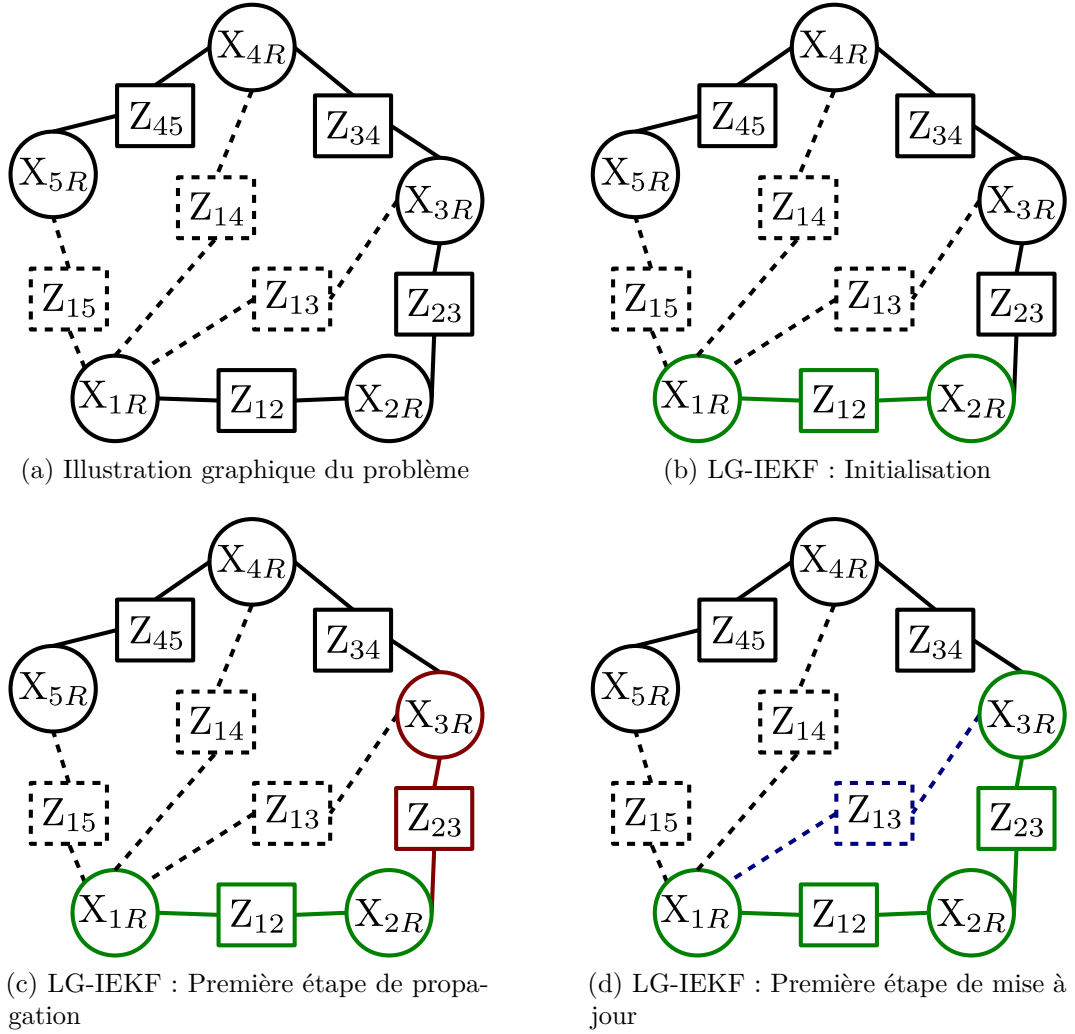


Figure 3.1 – Illustration graphique du problème de débruitage de transformations relatives : les données en pointillés (Z_{ij} où $j > i + 1$) sont utilisées comme observations durant l'étape de mise à jour du LG-IEKF alors que les transformations observées entre deux référentiels temporellement consécutifs ($Z_{(i+1)i}$) sont utilisées dans l'étape de propagation du LG-IEKF.

- Débruitage de transformations rigides relatives (groupe de Lie $SE(3)$), également connu sous le nom d'alignement de poses de caméra. Dans [Agrawal 2006], une transformation rigide est paramétrée par sa représentation vectorielle dans l'algèbre de Lie $\mathfrak{se}(3)$ de $SE(3)$ afin d'obtenir un problème non-linéaire euclidien sans contrainte extérieure. Cependant, cette paramétrisation crée des problèmes comme expliqué dans la partie 2.4.7.5. [Govindu 2004] propose un algorithme itératif qui peut être interprété comme un Gauss-Newton intrinsèque sur groupe de Lie dédié à $SE(3)$. Néanmoins, le critère minimisé n'est pas invariant (que ce soit par rapport à l'action à droite ou l'action à gauche du groupe de Lie sur lui-même) alors que le modèle d'observation l'est. Nous verrons que cette non prise en compte de la symétrie du problème conduit à une détérioration des résultats.
- Débruitage d'homographies relatives (groupe de Lie $SL(3)$), également connu sous le nom de mosaïque d'images. Dans [Caballero 2007], un EKF est proposé, prenant en compte la géométrie du groupe de Lie de manière extrinsèque en effectuant une reprojection à chaque itération. [Meidow 2011] propose un algorithme itératif basé sur une mise à jour des paramètres utilisant l'exponentielle de matrice. Cependant, ils ne tirent pas complètement partie de la structure du groupe de Lie $SL(3)$. Afin d'obtenir une solution efficace, [Khurd 2012] effectue une “relaxation” en ignorant les contraintes associées à $SL(3)$.

Obtention des équations du LG-IEKF Afin d'appliquer le LG-IEKF au problème du débruitage de transformations relatives, nous distinguons les transformations observées entre deux référentiels temporellement consécutifs ($Z_{(i+1)i}$ pour $i = 1 \dots N - 1$), que nous utilisons dans l'étape de propagation du filtre, des autres transformations relatives observées que nous utilisons dans l'étape de mise à jour du filtre. Nous spécifions maintenant les modèles de propagation et d'observation puis calculons les expressions de F_k et H_l (voir Alg.3.2).

Propagation L'état que nous cherchons à estimer contient les transformations absolues. Lors de l'initialisation, l'état contient les deux premières transformations absolues X_{1R} et X_{2R} . L'étape de propagation consiste à augmenter l'état avec la dernière transformation absolue X_{kR} déjà présente dans l'état (X_{2R} lors de la première étape de propagation) et à la propager en utilisant la transformation observée $Z_{(k+1)k}$. Cela revient à considérer le modèle

de propagation suivant :

$$\begin{aligned} X_k &= \exp_{G_k}^{\wedge} \left(\begin{bmatrix} w_k \\ 0 \end{bmatrix} \right) \begin{bmatrix} Z_{(k+1)k} (X_{k-1})_{kR} \\ X_{k-1} \end{bmatrix} \\ &= \exp_{G_k}^{\wedge} \left(\begin{bmatrix} w_k \\ 0 \end{bmatrix} \right) \begin{bmatrix} Z_{(k+1)k} & \mathbf{0} \\ \mathbf{0} & Id_{p_{k-1}} \end{bmatrix} \begin{bmatrix} (X_{k-1})_{kR} & \mathbf{0} \\ \mathbf{0} & X_{k-1} \end{bmatrix} \end{aligned} \quad (3.79)$$

où $w_k = b_{(k+1)k}^{k+1}$.

Remarquons que dans notre modèle, le groupe de Lie sur lequel évoluent les paramètres change à chaque propagation car l'état augmente contrairement à (3.27), c'est à dire qu'à l'instant k , le groupe de Lie des paramètres G_k (de dimension p_k) a la forme suivante : $G_k = G \times G \times \dots \times G$, i.e $k+1$ produits directs de G . En pratique, le fait que le groupe de Lie évolue à chaque propagation ne change rien vis à vis de l'applicabilité du LG-IEKF.

Afin d'appliquer la propagation du LG-IEKF, il faut calculer l'expression de F_k (3.39) :

$$\begin{aligned} &\log_{G_k}^{\vee} \left(f(\mu_{k-1|k-1}) f(\exp_{G_{k-1}}^{\wedge}(s) \mu_{k-1|k-1})^{-1} \right) \\ &= \log_{G_k}^{\vee} \left(\begin{bmatrix} Z_{(k+1)k} & \mathbf{0} \\ \mathbf{0} & Id_{p_{k-1}} \end{bmatrix} \begin{bmatrix} (\mu_{k-1|k-1})_{kR} & \mathbf{0} \\ \mathbf{0} & \mu_{k-1|k-1} \end{bmatrix} \right) \\ &\quad \left(\begin{bmatrix} Z_{(k+1)k} & \mathbf{0} \\ \mathbf{0} & Id_{p_{k-1}} \end{bmatrix} \exp_{G_k}^{\wedge} \left(\begin{bmatrix} \delta_{kR} \\ \delta \end{bmatrix} \right) \begin{bmatrix} (\mu_{k-1|k-1})_{kR} & \mathbf{0} \\ \mathbf{0} & \mu_{k-1|k-1} \end{bmatrix} \right)^{-1} \\ &= \log_{G_k}^{\vee} \left(\begin{bmatrix} Z_{(k+1)k} & \mathbf{0} \\ \mathbf{0} & Id_{p_{k-1}} \end{bmatrix} \exp_{G_k}^{\wedge} \left(- \begin{bmatrix} \mathbf{0} & Id_q & \mathbf{0} \\ Id_{p_{k-1}} \end{bmatrix} \delta \right) \begin{bmatrix} Z_{(k+1)k}^{-1} & \mathbf{0} \\ \mathbf{0} & Id_{p_{k-1}} \end{bmatrix} \right) \\ &= - \begin{bmatrix} \text{Ad}_{G'}(Z_{(k+1)k}) & \mathbf{0} \\ \mathbf{0} & Id_{p_{k-1}} \end{bmatrix} \begin{bmatrix} \mathbf{0} & Id_q & \mathbf{0} \\ Id_{p_{k-1}} \end{bmatrix} \delta \end{aligned} \quad (3.81)$$

Ainsi

$$F_k = \begin{bmatrix} \text{Ad}_{G'}(Z_{(k+1)k}) & \mathbf{0} \\ \mathbf{0} & Id_{p_{k-1}} \end{bmatrix} \begin{bmatrix} \mathbf{0} & Id_q & \mathbf{0} \\ Id_{p_{k-1}} \end{bmatrix} \quad (3.82)$$

Mise à jour Le modèle de mise à jour que nous considérons correspond à (3.78), c'est à dire :

$$Z_{ij} = \exp_{G'}^{\wedge}(b_{ij}^i) (X_k)_{iR} (X_k)_{jR}^{-1} \quad (3.83)$$

où $b_{ij}^i \sim \mathcal{N}_{\mathbb{R}^q}(\mathbf{0}_{q \times 1}, \Sigma_{ij}^i)$ est un bruit blanc gaussien.

Afin d'appliquer la mise à jour du LG-IEKF, il faut calculer l'expression de

H_l (3.53) :

$$\begin{aligned}
 & \log_{G'}^\vee \left(Z_{ij} \exp_{G'}^\wedge (\delta_{jR}) X_{jR} X_{iR}^{-1} \exp_{G'}^\wedge (-\delta_{iR}) \right) \\
 &= \log_{G'}^\vee \left(Z_{ij} X_{jR} X_{iR}^{-1} \exp_{G'}^\wedge \left(\text{Ad}_{G'} \left(X_{iR} X_{jR}^{-1} \right) \delta_{jR} \right) \exp_{G'}^\wedge (-\delta_{iR}) \right) \\
 &\simeq \log_{G'}^\vee \left(Z_{ij} X_{jR} X_{iR}^{-1} \right) - \delta_{iR} + \text{Ad}_{G'} \left(X_{iR} X_{jR}^{-1} \right) \delta_{jR}
 \end{aligned} \tag{3.84}$$

Ainsi

$$H_l = \begin{bmatrix} \mathbf{0} & Id_q & \mathbf{0} & -\text{Ad}_{G'} \left(X_{iR}^l \left(X_{jR}^l \right)^{-1} \right) & \mathbf{0} \end{bmatrix} \tag{3.85}$$

Une illustration graphique des étapes de propagation et mise à jour sont présentées en figure 3.1.

Dans notre implémentation, φ_l et Φ_l sont approchés par la matrice identité. Remarquons également que le fait que le modèle d'observation soit invariant vis à vis de l'action du groupe sur lui-même induit que la matrice de covariance de l'état est dégénérée, c'est à dire que ses p plus petites valeurs propres sont nulles. En pratique, cela ne crée pas d'instabilité numérique car dans ce cas, il est possible de montrer que lors de l'étape de mise à jour, la correction apportée par le LG-IEKF se fait de manière orthogonale aux directions dégénérées.

Résultats sur données simulées pour un problème d'alignement de poses de caméra Nous avons choisi de comparer le LG-IEKF aux algorithmes de l'état de l'art sur un problème d'alignement de poses de caméra. Nous considérons 6 algorithmes différents :

- “Chaîne” : simple composition des transformations observées entre deux référentiels temporellement consécutifs
- “Euc-GN-LieAlg” : algorithme proposé dans [Agrawal 2006] (Agrawal et al.)
- “LG-GN-Non-Invariant” : algorithme proposé dans [Govindu 2004] (Govindu)
- “LG-GN” : algorithme LG-GN présenté dans la partie 3.4 où l'on minimise

$$\sum_{i,j} \left\| \log_{G'}^\vee \left(Z_{ij} X_{jR} X_{iR}^{-1} \right) \right\|_{\Sigma_{ij}^i}^2 \tag{3.86}$$

- “LG-IEKF” : algorithme LG-IEKF Alg.3.2
- “LG-EKF” : algorithme LG-IEKF Alg.3.2 où une seule itération lors de l'étape de mise à jour est effectuée

Nous simulons des trajectoires circulaires (voir Fig.3.3) comportant N poses de caméra. Les transformations relatives bruitées sont obtenues en utilisant le modèle génératif (3.78).

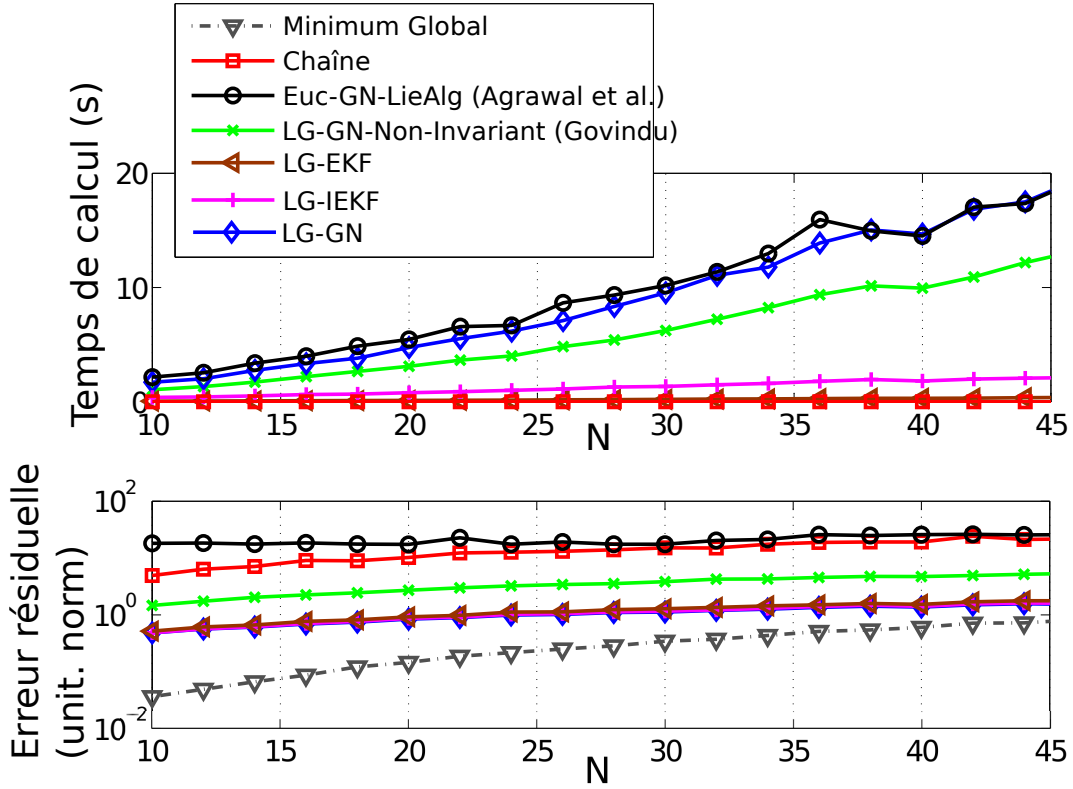


Figure 3.2 – Temps de calcul et erreur résiduelle en fonction du nombre de transformations absolues à estimer (N). Le nombre de transformations relatives observées est fixé à $N + n$ (pour cette expérience $n = 30$).

Afin de ne pas avantager le LG-IEKF par rapport aux autres algorithmes ([Govindu 2004], [Agrawal 2006] et LG-GN), ces derniers sont employés de manière incrémentale sur les données, ce qui permet de guider l'estimation et réduit les chances de tomber dans un mauvais minimum local.

Tous les algorithmes ont été codés en Matlab et testés sur la configuration suivante : core I5 4x2.27GHz, 4GB, Linux 64 bits.

Finalement, afin de comparer les résultats obtenus à la trajectoire simulée de la caméra, nous devons aligner les estimés des poses des caméras avec les vraies poses simulées. Pour cela, nous employons un LG-GN dans le but de minimiser la somme des erreurs suivantes :

$$\left\| \log_{SE(3)}^{\vee} \left(\mu_{iR} X_{RR_{True}} X_{iR_{True}}^{-1} \right) \right\|^2 \quad (3.87)$$

L'erreur finale obtenue à l'issue de ce LG-GN pour chaque méthode ainsi que les temps de calculs sont présentés en figure 3.2.

Les résultats montrent que le LG-IEKF produit des résultats sensiblement meilleurs que ceux des algorithmes de l'état de l'art [Agrawal 2006] et [Govindu 2004].

En effet, la paramétrisation choisie dans [Agrawal 2006] crée des minima locaux alors que l'approche de [Govindu 2004] qui ne tire pas complètement partie de la géométrie du problème (l'erreur considérée n'étant pas invariante) tombe fréquemment dans un mauvais minimum local.

Comme nous nous y attendions, le LG-GN qui conserve toutes les données au cours du temps produit des performances légèrement meilleures que le LG-IEKF qui “résume” l'information passée par une distribution normale concentrée à chaque pas de temps. Cependant, le LG-IEKF a un temps de calcul très inférieur à celui du LG-GN.

Finalement, le fait de ne pas itérer l'étape de mise à jour (LG-EKF) dégrade légèrement les performances mais permet à nouveau un gain en termes de temps de calcul.

3.5.7.3 Cas 2 : Estimation en présence de données aberrantes

Nous considérons maintenant le cas où les observations contiennent des données aberrantes et comparons les performances du LG-IEKF, muni du fenêtrage statistique avant chaque étape de mise à jour, aux algorithmes de l'état de l'art capables de prendre en compte la présence de données aberrantes.

Comme nous le verrons dans les différentes expériences, les données aberrantes proviennent le plus souvent de structures identiques apparaissant plusieurs fois dans l'environnement [Roberts 2011].

État de l'art Plusieurs approches dédiées au débruitage de rotations relatives en présence de données aberrantes ont été proposées. Ce problème est appelé synchronisation de rotations par la communauté mathématique et les solutions proposées consistent souvent à minimiser un critère.

Dans [Bandeira 2012] et [Singer 2011], des relaxations spectrales du problème sont proposées alors que [Boumal 2013b] utilise le résultat de ces approches comme initialisation pour un algorithme d'optimisation de type région de confiance du deuxième ordre sur variété Riemannienne.

[Wang 2013] présente un algorithme capable de retrouver exactement les rotations absolues, lorsqu'une partie des observations ne sont pas bruitées. Cet algorithme obtient de meilleures performances que [Singer 2011]. Dans [Hartley 2011] et [Chatterjee 2013], deux approches itératives robustes visant à minimiser des critères basés respectivement sur la norme L1 et la norme L1-L2 sont proposés. Cependant, les fonctions minimisées possèdent plusieurs minima locaux et nécessitent donc une “bonne” initialisation.

Finalement, [Crandall 2011] propose une discrétisation de $SO(3)$ et applique un algorithme de type “Loopy Belief Propagation” (LBP).

Les travaux de [Enqvist 2011], [Jiang] et [Moulon 2013] traitent également du problème de débruitage de rotations relatives. Cependant, ces approches sont spécifiquement dédiées à la topologie de $SO(3)$ et leur généralisation à d'autres groupes de Lie n'est pas directe.

A notre connaissance, une seule approche [Roberts 2011], permettant de traiter du problème générique du débruitage de transformations relatives en présence de données aberrantes, a été développée. Il s'agit d'un algorithme de type "Espérance Maximisation" (EM) où des variables cachées, permettant de classer chaque donnée comme aberrante ou non, sont introduites.

Obtention des équations du LG-IEKF Afin d'appliquer le LG-IEKF, nous faisons l'hypothèse que les transformations observées entre deux référentiels temporellement consécutifs ($Z_{(i+1)i}$ pour $i = 1 \dots N - 1$) ne contiennent pas de données aberrantes.

Cette hypothèse peut paraître restrictive. Cependant, pour une séquence d'images par exemple, elle est généralement vérifiée (voir partie 3.5.7.3 et 3.5.7.3).

Le LG-IEKF que nous employons dans cette partie est le même que celui utilisé dans la partie 3.5.7.2. Néanmoins, cette fois, nous effectuons un fenêtrage statistique, avec un seuil ayant une valeur égale à 99.9% de la p-valeur du χ^2 , avant chaque étape de mise à jour pour supprimer les données aberrantes.

Résultats sur données simulées pour un problème d'alignement de poses de caméra Nous comparons maintenant les performances du LG-IEKF (avec fenêtrage statistique) par rapport à celles des deux algorithmes de l'état de l'art [Roberts 2011] et [Chatterjee 2013] sur un problème d'alignement de poses de caméra (groupe de Lie $SE(3)$).

L'approche de [Chatterjee 2013] est dédiée à $SO(3)$ mais son extension à $SE(3)$ est directe.

Nous simulons des trajectoires circulaires (voir Fig.3.3) comportant N poses de caméra. Chaque pose $X_{iR_{True}}$ possède également un horodatage t_i qui nous permet de générer les observations de la façon suivante :

Tout d'abord, une observation peut être soit une donnée aberrante soit une donnée non aberrante (sauf dans le cas où les deux référentiels sont temporellement consécutifs auquel cas, par hypothèse, elle est obligatoirement non aberrante). Nous modélisons la probabilité qu'une observation ne soit pas aberrante comme :

$$P(Z_{ij} \text{ non aberrant}) = \exp(-\lambda |t_i - t_j|) \quad (3.88)$$

où λ est un paramètre à fixer. Ce modèle signifie simplement qu'un grand écart temporel augmente la chance de produire une donnée aberrante. Après avoir tiré la classe d'une observation (aberrante ou non), nous tirons un échantillon selon l'équation (3.78) pour une donnée non aberrante alors que la distribution d'une donnée aberrante est modélisée par une distribution normale concentrée sur groupe de Lie centrée avec une grande covariance.

Dans la figure 3.4, nous comparons les résultats du LG-IEKF (avec fenêtrage statistique) aux approches robustes de [Chatterjee 2013] et l'algorithme EM de [Roberts 2011].

Les algorithmes [Chatterjee 2013] et [Roberts 2011] sont initialisés en composant les transformations relatives temporellement consécutives comme proposé dans leurs articles. Les paramètres de ces algorithmes ont été optimisés à la main, mais les résultats obtenus étaient en pratique peu sensible aux changements de ces paramètres.

Comme dans la partie 3.5.7.2, afin d'aligner les résultats avec la vérité terrain, nous utilisons un LG-GN. Les erreurs résiduelles obtenues pour chaque approche sont présentées en figure 3.4.

Nous pouvons voir que le LG-IEKF produit à la fois de meilleurs résultats que [Chatterjee 2013] et que [Roberts 2011].

En effet, l'approche de [Chatterjee 2013] est basée sur une norme L1-L2 qui permet de réduire l'influence des données aberrantes. Cependant, le problème minimisé possède des minima locaux et l'algorithme reste régulièrement bloqué dans un "mauvais" minimum local.

L'approche de [Roberts 2011] introduit des variables cachées permettant de classer chaque observation comme une donnée aberrante ou non. Cependant, la classification obtenue à l'initialisation de l'algorithme est difficile à modifier au cours des itérations car, dans ce formalisme, l'étape E de l'EM ne prend pas en compte les erreurs d'estimation qui sont uniquement négligeables lorsque N est petit. Par conséquent, beaucoup de données non aberrantes restent classifiées comme aberrantes.

Contrairement à ces approches, notre algorithme classe les observations de manière incrémentale tout en prenant en compte les erreurs d'estimation par le biais de la matrice de covariance qui est également estimée.

Un exemple de résultat obtenu avec les trois différentes approches est présenté en figure 3.3.

Résultats sur données réelles pour un problème d'alignement de modèles 3D partiels Nous comparons maintenant les performances du LG-IEKF (avec fenêtrage statistique) par rapport à celles des deux algorithmes de l'état de l'art [Roberts 2011] et [Chatterjee 2013] sur un problème d'alignement

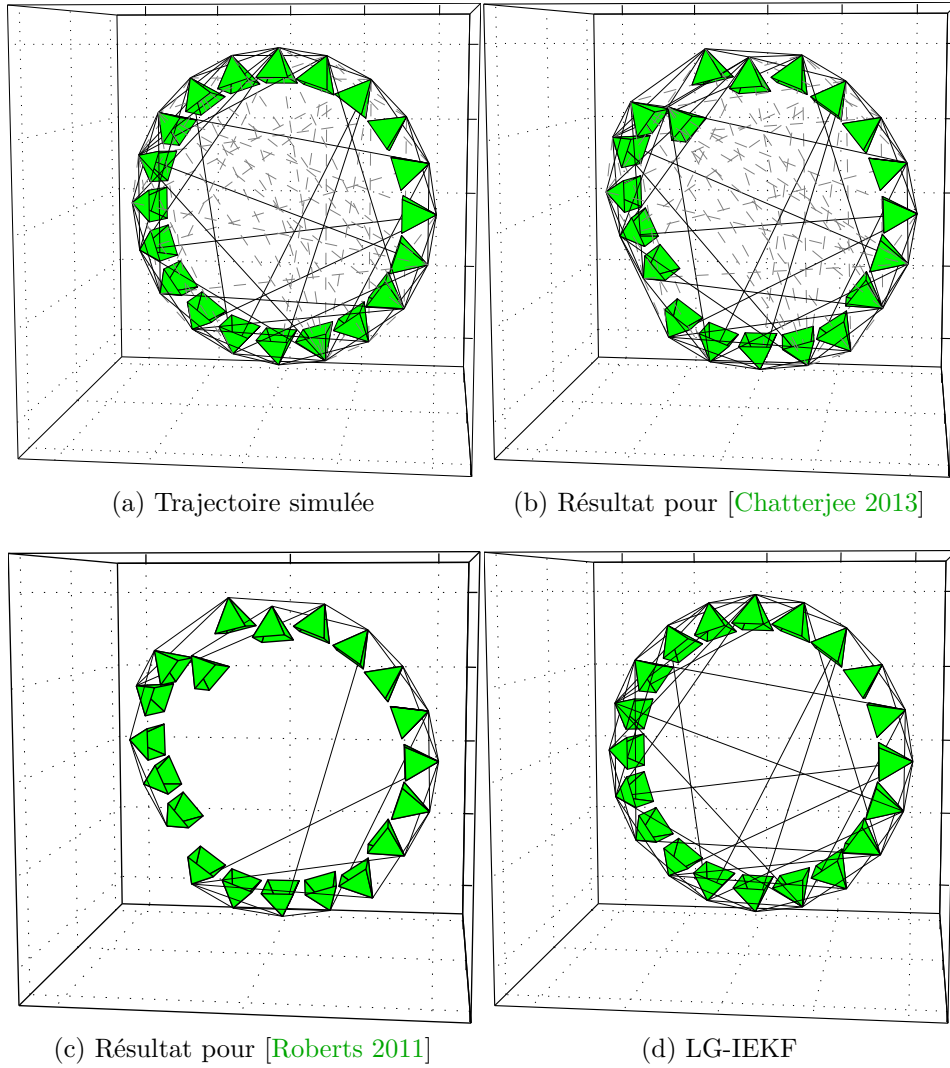


Figure 3.3 – Exemple de résultat pour un problème d’alignement de poses de caméra. Un cône représente la pose d’une caméra, une ligne noire indique la présence d’une observation non aberrante entre deux poses alors qu’une ligne grise en pointillés indique la présence d’une observation aberrante entre les deux poses.

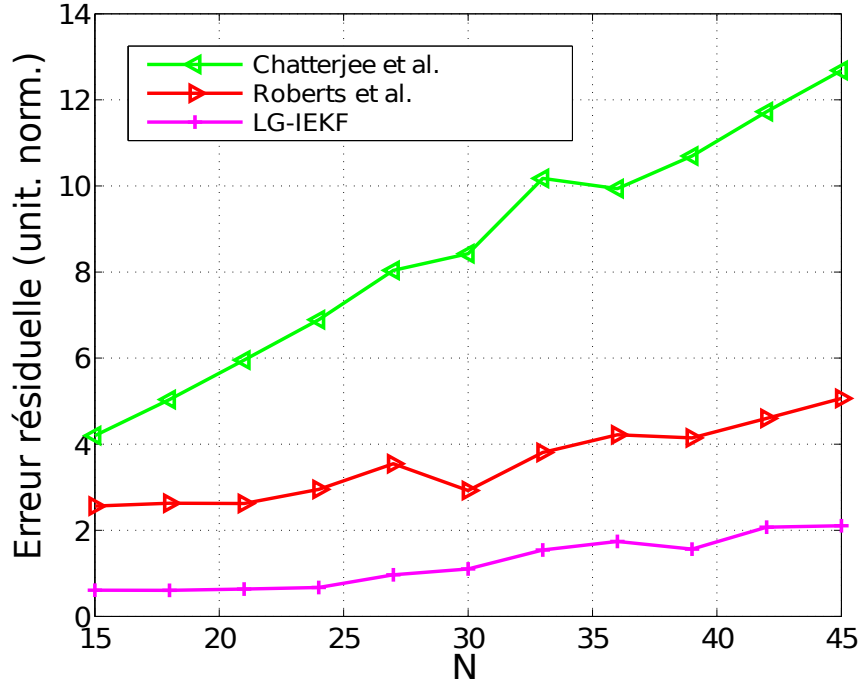


Figure 3.4 – Erreurs résiduelles en fonction du nombre de transformations absolues à estimer (N) pour lequel $\lambda = \frac{1}{10}$ et le nombre de transformations relatives observées est fixé à $5N + n$ avec $n = 60$.

de modèles 3D partiels (groupe de Lie $Sim(3)$). Les conditions de l'expérience sont les suivantes : la caméra évolue autour de deux objets identiques qui sont dupliqués dans l'environnement. L'objectif ici est d'estimer la trajectoire de la caméra le plus précisément possible.

Afin d'estimer la trajectoire, nous commençons par couper la vidéo en morceaux avec un facteur de recouvrement $1/2$. Pour chaque morceau, nous utilisons un algorithme de SLAM similaire à [Klein 2007] qui estime un nuage de points 3D ainsi que les poses des caméras. Ensuite, pour chaque paire de nuages de points 3D, nous estimons une similitude 3D avec un algorithme de RANSAC suivi d'un LG-GN ce qui fournit une transformation relative que nous allons utiliser comme observation dans le LG-IEKF. La covariance de l'observation est obtenue en effectuant une approximation de Gauss-Laplace.

La difficulté du problème provient du fait que deux objets identiques sont présents dans la scène. Par conséquent, à chaque fois que nous essayons d'aligner le nuage de points représentant une partie d'un objet avec un nuage de points représentant l'autre objet, cet alignement semble valide ce qui produit une donnée aberrante.

Une fois les transformations absolues estimées, il s'agit d'aligner les différents morceaux de trajectoire issus de l'algorithme de SLAM pour obtenir une trajectoire globale.

Dans le but de comparer les résultats quantitativement, nous avons manuel-

lement annoté chaque observation comme étant une donnée aberrante ou non. Afin de juger de la qualité de la reconstruction, nous avons fait en sorte que la pose de la caméra soit la même au début et à la fin de la vidéo.

Dans la figure 3.5, nous comparons les résultats obtenus avec les trois approches différentes.

D'un côté, nous remarquons qu'à la fois [Roberts 2011] et [Chatterjee 2013] n'ont pas été capables d'estimer correctement les transformations globales car, pour ces deux méthodes, la première et la dernière pose sont loin d'être superposées. De plus, nous pouvons voir que l'approche de [Roberts 2011] a classifié de nombreuses données non aberrantes comme aberrantes. D'un autre côté, le LG-IEKF muni du fenêtrage statistique a parfaitement classifié les données aberrantes et les données non aberrantes (voir Fig.3.5e). De plus, la première et la dernière caméra sont presque parfaitement superposées.

Résultats sur données réelles pour un problème de mosaïque d'images

Nous comparons maintenant les performances du LG-IEKF (avec fenêtrage statistique) par rapport à celles des deux algorithmes de l'état de l'art [Roberts 2011] et [Chatterjee 2013] sur un problème de mosaïque d'images (groupe de Lie $SL(3)$).

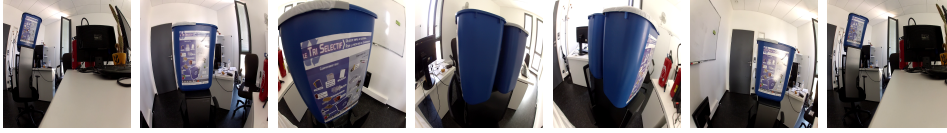
Pour cela, nous avons pris 53 photos de feuilles de papier étalées sur le sol formant un carré (voir Fig.3.6a) avec un smartphone. Ensuite nous avons détecté et apparié des points d'intérêt et calculé une homographie entre toutes les paires d'images à l'aide d'un algorithme de RANSAC suivi d'un LG-GN ce qui nous donne des transformations relatives que nous allons utiliser comme observations dans notre LG-IEKF. Les covariances des observations sont obtenues en effectuant une approximation de Gauss-Laplace. Certaines feuilles de papier sont dupliquées ce qui donne lieu à de nombreuses transformations relatives aberrantes. Pour ce jeu de données, il y a 65% de données aberrantes (voir Fig.3.6b).

Remarquons que pour ce jeu de données, la transformation relative entre l'image 43 et l'image 44 n'est pas disponible. A la place, nous utilisons la transformation relative entre l'image 43 et l'image 45 lors de l'étape de propagation du LG-IEKF.

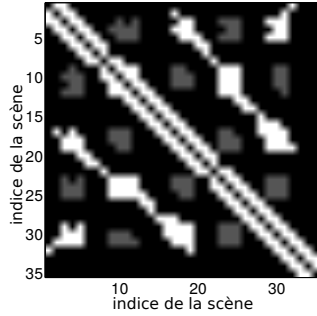
Dans la figure 3.6, nous comparons les résultats obtenus avec les trois approches différentes.

D'un côté, l'approche de [Roberts 2011] classifie à nouveau de nombreuses données non aberrantes comme aberrantes puisque les erreurs d'estimation de sont pas prises en compte lors de l'étape E de l'EM. Par conséquent, les transformations absolues sont mal estimées (voir Fig.3.6c).

D'un autre côté, le LG-IEKF muni du fenêtrage statistique a parfaitement



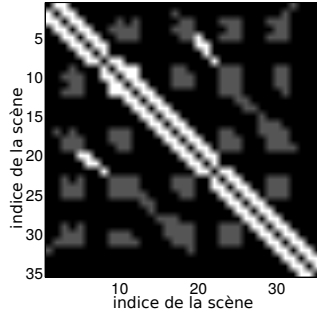
(a) Exemple d'images de la séquence vidéo. La pose de la caméra au début et à la fin de la vidéo est la même.



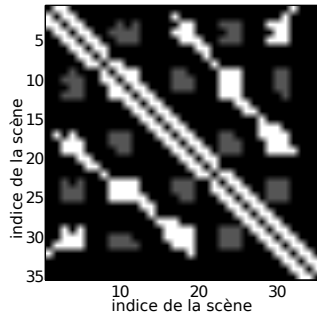
(b) Vérité terrain : matrice de classification des données aberrantes annotée manuellement



(c) [Chatterjee 2013] : trajectoire estimée



(d) [Roberts 2011] : (gauche) matrice de classification des données aberrantes, (droite) trajectoire estimée



(e) LG-IEKF : (gauche) matrice de classification des données aberrantes, (droite) trajectoire estimée

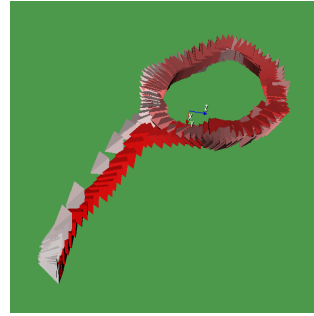
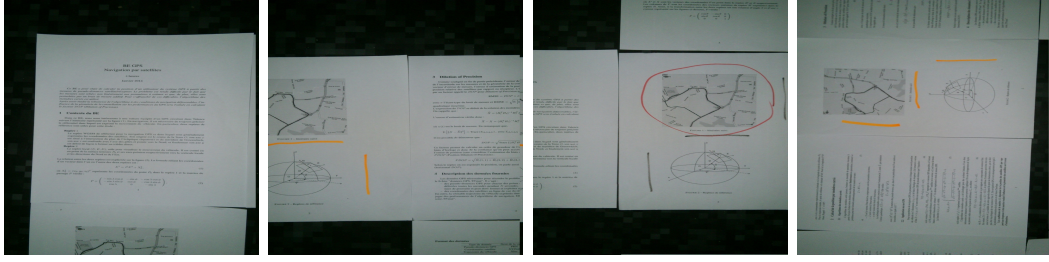
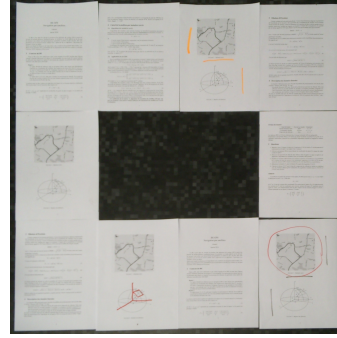
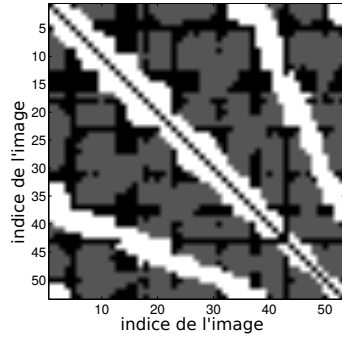


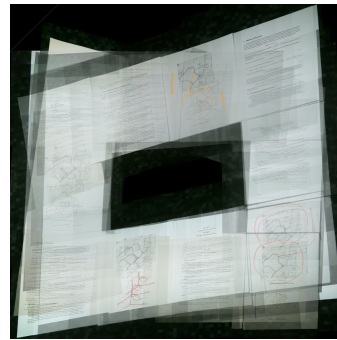
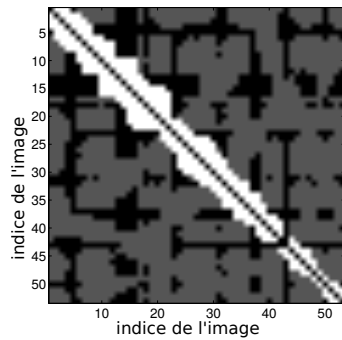
Figure 3.5 – Résultats pour le problème d'alignement de modèles 3D partiels : dans une matrice de classification des données aberrantes, un pixel blanc indique une donnée non aberrante, un pixel gris indique une donnée aberrante et un pixel noir indique qu'il n'y a pas d'observation. Concernant la trajectoire estimée de la caméra, la pose de la caméra pour 1ère image de la vidéo est représentée par un cône rouge foncé, alors que la dernière est représentée par un cône rouge clair. Dans la trajectoire réelle, la pose de la caméra au début et à la fin de la vidéo est la même. En pratique, la première et la dernière pose estimée devraient donc parfaitement se superposer.



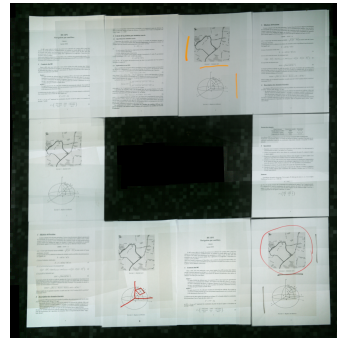
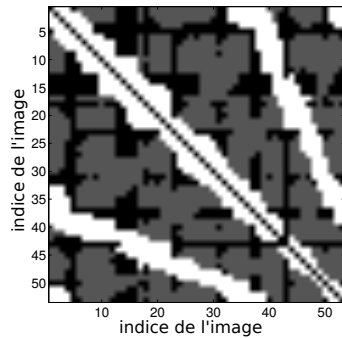
(a) Exemples d'images à partir desquelles nous voulons créer la mosaïque



(b) Vérité terrain : (gauche) matrice de classification des données aberrantes annotée manuellement, (droite) vue de dessus de la scène



(c) [Roberts 2011] : (gauche) matrice de classification des données aberrantes, (droite) mosaïque



(d) LG-IEKF : (gauche) matrice de classification des données aberrantes, (droite) mosaïque

Figure 3.6 – Résultats pour le problème de mosaïque d'images : dans une matrice de classification des données aberrantes, un pixel blanc indique une donnée non aberrante, un pixel gris indique une donnée aberrante et un pixel noir indique simplement qu'il n'y a pas d'observation.

classifié les données aberrantes et les données non aberrantes et produit une mosaïque visuellement très proche de la vérité terrain (voir Fig.3.6d).

L’approche de [Chatterjee 2013] n’a pas pu être employée sur ce problème pour cause d’instabilité numérique.

3.6 Lisseur de Rauch–Tung–Striebel

Le lisseur de Rauch–Tung–Striebel est considéré comme une méthode de référence dans le domaine de l’estimation non linéaire de paramètres euclidiens à partir d’observations elles-mêmes euclidiennes. Ce lisseur est construit sur la sortie d’un filtre de Kalman, c’est à dire qu’il permet de corriger les estimés issus du filtre de Kalman en incluant l’information contenue dans les observations venant du futur.

Ainsi, pour appliquer ce lisseur, il faut tout d’abord parcourir les données dans le sens “direct”, c’est à dire de l’instant $k = 1$ à l’instant $k = T$ (où T est la durée de la fenêtre d’observation), en leur appliquant un filtre de Kalman. Le lisseur, qui fonctionne à rebours, c’est à dire de l’instant $k = T$ à l’instant $k = 1$, peut alors être appliqué. Par conséquent, à l’issue du lisseur, l’estimé de l’état au temps k ne tient pas seulement compte des observations passées z_1, \dots, z_k mais de toutes les observations contenues dans la fenêtre d’observation z_1, \dots, z_T .

Dans ce sous-chapitre, nous proposons d’étendre cette approche au cas où l’état et les observations évoluent sur des groupes de Lie, en utilisant les outils présentés dans les sous-chapitres précédents, à savoir l’approximation de Gauss-Laplace intrinsèque et l’algorithme d’optimisation de Gauss-Newton intrinsèque.

3.6.1 Rappels sur le lisseur de Rauch–Tung–Striebel sur espace euclidien

Le lisseur de Kalman étendu, également connu sous le nom de lisseur de Rauch–Tung–Striebel étendu (RTS) [Särkkä 2013] est un lisseur dédié aux systèmes non linéaires. L’objectif de ce lisseur est de récursivement approcher la distribution a posteriori $p(x_k | z_1, \dots, z_T)$, par une distribution normale multivariée, où $x_k \in \mathbb{R}^p$ est l’état que nous souhaitons estimer à l’instant k et $z_l \in \mathbb{R}^q$ est l’observation disponible à l’instant l . Pour ce faire, la forme des

distributions suivante est supposée :

$$p(x_{k+1}, x_k | z_{1:T}) = \mathcal{N}_{\mathbb{R}^{2p}} \left(\begin{bmatrix} x_{k+1} \\ x_k \end{bmatrix}; \begin{bmatrix} \mu_{k+1|k} \\ \mu_{k|k} \end{bmatrix}, \begin{bmatrix} P_{k+1|k} = R_k + F_k P_{k|k} F_k^T & F_k P_{k|k} \\ P_{k|k} F_k^T & P_{k|k} \end{bmatrix} \right) \quad (3.89)$$

$$p(x_{k+1} | z_{1:k}) = \mathcal{N}_{\mathbb{R}^p} (x_{k+1}; \mu_{k+1|k}, P_{k+1|k} = R_k + F_k P_{k|k} F_k^T) \quad (3.90)$$

qui correspondent à la sortie de l'étape de propagation de l'IEKF (ou de l'EKF) respectivement avant et après marginalisation. Nous supposons également connaître la sortie du lisseur à l'instant $k + 1$ (le lisseur marche à rebours, c'est à dire qu'il part de l'instant T pour arriver à l'instant 1) :

$$p(x_{k+1} | z_{1:T}) = \mathcal{N}_{\mathbb{R}^p} (x_{k+1}; \mu_{k+1|T}, P_{k+1|T}) \quad (3.91)$$

Par conséquent, à l'instant $k = T - 1$, le lisseur utilise la sortie de l'étape de mise à jour de l'IEKF (ou de l'EKF) qui a la forme :

$$p(x_T | z_{1:T}) = \mathcal{N}_{\mathbb{R}^p} (x_T; \mu_{T|T}, P_{T|T}) \quad (3.92)$$

Le Lisseur RTS étant fondé sur la sortie d'un filtre de Kalman, il suppose les mêmes modèles de propagation et de mise à jour que ce dernier (voir paragraphe 3.5.1).

Alors que la manière classique d'obtenir les équations du RTS est de linéariser le modèle de propagation [Särkkä 2013], il est possible d'utiliser une approximation de Gauss-Laplace en se fondant sur les propriétés de Markov du modèle :

$$\begin{aligned} p(x_{k+1}, x_k | z_{1:T}) &= p(x_k | x_{k+1}, z_{1:T}) p(x_{k+1} | z_{1:T}) \\ &= p(x_k | x_{k+1}, z_{1:k}) p(x_{k+1} | z_{1:T}) \\ &= \frac{p(x_{k+1}, x_k | z_{1:k})}{p(x_{k+1} | z_{1:k})} p(x_{k+1} | z_{1:T}) \end{aligned} \quad (3.93)$$

Ainsi en utilisant un algorithme de Gauss-Newton pour trouver le mode de (3.93), suivi d'une approximation de Gauss-Laplace pour la covariance et d'une marginalisation de x_{k+1} , il est possible d'aboutir aux expressions du lisseur.

3.6.1.1 Algorithme RTS

L'algorithme RTS est présenté ci-après :

Algorithme 3.3 RTS

Entrée : $\mu_{k|k}, P_{k|k}, \mu_{k+1|k}, P_{k+1|k}, \mu_{k+1|T}, P_{k+1|T}$

Sortie : $\mu_{k|T}, P_{k|T}$

$$\begin{aligned} L_k &= P_{k|k} F_k^T P_{k+1|k}^{-1} \\ \mu_{k|T} &= \mu_{k|k} + \left(L_k \left(\mu_{k+1|T} - \mu_{k+1|k} \right) \right) \\ P_{k|T} &= P_{k|k} + L_k \left(P_{k+1|T} - P_{k+1|k} \right) L_k^T \end{aligned}$$

Cet algorithme, qui est considéré comme une méthode de référence dans le domaine de l'estimation non linéaire, est restreint au cas où l'état et les observations évoluent sur des espaces euclidiens. Dans la suite de ce paragraphe, nous proposons d'étendre cette approche au cas où l'état et les observations évoluent sur des groupes de Lie.

3.6.2 Modèle proposé

Afin de généraliser le lisseur RTS aux groupes de Lie, il convient de généraliser les hypothèses du lisseur euclidien afin de prendre en compte le fait que l'état évolue sur un groupe de Lie.

Ainsi, nous proposons d'approcher récursivement la distribution a posteriori $p(X_{k+1}, X_k | Z_{1:T})$, non plus par une distribution normale multivariée, mais par une distribution normale concentrée sur groupe de Lie, où $X_k \in G$ est l'état que nous souhaitons estimer à l'instant k et $Z_l \in G'$ est l'observation disponible à l'instant l .

De plus, au lieu d'être fondée sur la sortie d'un filtre de Kalman euclidien (IEKF ou EKF), le lisseur que nous développons utilise la sortie du LG-IEKF (ou du LG-EKF) présenté dans la première partie de ce chapitre.

Les propriétés de Markov sont les mêmes que pour le LG-IEKF (voir paragraphe 3.5.2), à savoir un modèle markovien sur l'état $X_k \in G$, l'observation Z_k dépend uniquement de l'état courant X_k et est conditionnellement indépendante des autres états sachant l'état X_k . Ainsi, nous avons :

$$\begin{aligned} p(X_{k+1}, X_k | Z_{1:T}) &= p(X_k | X_{k+1}, Z_{1:T}) p(X_{k+1} | Z_{1:T}) \\ &= p(X_k | X_{k+1}, Z_{1:k}) p(X_{k+1} | Z_{1:T}) \\ &= \frac{p(X_{k+1}, X_k | Z_{1:k})}{p(X_{k+1} | Z_{1:k})} p(X_{k+1} | Z_{1:T}) \end{aligned} \quad (3.94)$$

Nous supposons que les trois distributions qui interviennent dans (3.94) ont la

forme suivante :

$$p(X_{k+1}, X_k | Z_{1:k}) = \mathcal{N}_{G \times G}^R \left(\begin{bmatrix} X_{k+1} \\ X_k \end{bmatrix}; \begin{bmatrix} \mu_{k+1|k} \\ \mu_{k|k} \end{bmatrix}, \begin{bmatrix} P_{k+1|k} & F_k P_{k|k} \\ P_{k|k} F_k^T & P_{k|k} \end{bmatrix} \right) \quad (3.95)$$

$$p(X_{k+1} | Z_{1:k}) = \mathcal{N}_G^R(X_{k+1}; \mu_{k+1|k}, P_{k+1|k} = R_k + F_k P_{k|k} F_k^T) \quad (3.96)$$

Ces deux premières distributions correspondent à la sortie de l'étape de propagation du LG-IEKF (ou LG-EKF) respectivement avant et après marginalisation. Nous supposons également connaître la sortie du lisseur à l'instant $k + 1$ (le lisseur marche à rebours, c'est à dire qu'il part de l'instant T pour arriver à l'instant 1) :

$$p(X_{k+1} | Z_{1:T}) = \mathcal{N}_G^R(X_{k+1}; \mu_{k+1|T}, P_{k+1|T}) \quad (3.97)$$

3.6.3 Solution proposée

Nous proposons d'ajuster une distribution normale concentrée sur groupe de Lie sur la densité de probabilité (3.94) à l'aide d'une approximation de Gauss-Laplace intrinsèque, suivi d'une marginalisation de X_{k+1} (voir annexe A). Sous l'hypothèse de Gaussienne concentrée, l'opposé du logarithme de (3.94) est :

$$\begin{aligned} l(X_{k+1}, X_k) &= \left\| \begin{bmatrix} \log_G^\vee(X_{k+1} \mu_{k+1|k}^{-1}) \\ \log_G^\vee(X_k \mu_{k|k}^{-1}) \end{bmatrix} \right\|^2 \\ &\quad - \left\| \log_G^\vee(X_{k+1} \mu_{k+1|k}^{-1}) \right\|_{P_{k+1|k}}^2 + \left\| \log_G^\vee(X_{k+1} \mu_{k+1|T}^{-1}) \right\|_{P_{k+1|T}}^2 \end{aligned} \quad (3.98)$$

$$= \left\| \begin{bmatrix} \log_G^\vee(X_{k+1} \mu_{k+1|k}^{-1}) \\ \log_G^\vee(X_k \mu_{k|k}^{-1}) \\ \log_G^\vee(X_{k+1} \mu_{k+1|k}^{-1}) \\ \log_G^\vee(X_{k+1} \mu_{k+1|T}^{-1}) \end{bmatrix} \right\|_{\mathfrak{E}}^2 \quad (3.99)$$

où

$$\mathfrak{E} = \begin{bmatrix} P_{k+1|k} & F_k P_{k|k} & \mathbf{0} & \mathbf{0} \\ P_{k|k} F_k^T & P_{k|k} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -P_{k+1|k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & P_{k+1|T} \end{bmatrix} \quad (3.100)$$

Afin de trouver $\{\hat{X}_{k+1}, \hat{X}_k\} = \underset{X_{k+1}, X_k}{\operatorname{argmin}} \{l(X_{k+1}, X_k)\}$, nous proposons d'appliquer un LG-GN.

3.6.4 Application du LG-GN

Nous commençons par linéariser le terme à l'intérieur de la norme de (3.99) en $X_k^l = X_k^0$ et $X_{k+1}^l = X_{k+1}^0$. Nous verrons par la suite, qu'il convient de prendre pour valeurs initiales $X_k^0 = \mu_{k|k}$ et $X_{k+1}^0 = \mu_{k+1|T}$.

Pour cela, nous définissons :

$$\delta_{k+1}^{l,k} = \log_G^\vee \left(X_{k+1}^l \mu_{k+1|k}^{-1} \right) \quad (3.101)$$

$$\delta_k^{l,k} = \log_G^\vee \left(X_k^l \mu_{k|k}^{-1} \right) \quad (3.102)$$

$$\delta_{k+1}^{l,T} = \log_G^\vee \left(X_{k+1}^l \mu_{k+1|T}^{-1} \right) \quad (3.103)$$

En utilisant (1.33) nous obtenons :

$$\log_G^\vee \left(\exp_G^\wedge \left(\delta_{k+1}^{l+1/l} \right) X_{k+1}^l \mu_{k+1|k}^{-1} \right) \simeq \delta_{k+1}^{l,k} + \varphi_G \left(\delta_{k+1}^{l,k} \right) \delta_{k+1}^{l+1/l} \quad (3.104)$$

$$\log_G^\vee \left(\exp_G^\wedge \left(\delta_k^{l+1/l} \right) X_k^l \mu_{k|k}^{-1} \right) \simeq \delta_k^{l,k} + \varphi_G \left(\delta_k^{l,k} \right) \delta_k^{l+1/l} \quad (3.105)$$

$$\log_G^\vee \left(\exp_G^\wedge \left(\delta_{k+1}^{l+1/l} \right) X_{k+1}^l \mu_{k+1|T}^{-1} \right) \simeq \delta_{k+1}^{l,T} + \varphi_G \left(\delta_{k+1}^{l,T} \right) \delta_{k+1}^{l+1/l} \quad (3.106)$$

Le nouveau problème que nous cherchons à résoudre s'écrit alors :

$$\left\| \begin{bmatrix} \delta_{k+1}^{l,k} + \varphi_G \left(\delta_{k+1}^{l,k} \right) \delta_{k+1}^{l+1/l} \\ \delta_k^{l,k} + \varphi_G \left(\delta_k^{l,k} \right) \delta_k^{l+1/l} \\ \delta_{k+1}^{l,k} + \varphi_G \left(\delta_{k+1}^{l,k} \right) \delta_{k+1}^{l+1/l} \\ \delta_{k+1}^{l,T} + \varphi_G \left(\delta_{k+1}^{l,T} \right) \delta_{k+1}^{l+1/l} \end{bmatrix} \right\|_{\mathfrak{E}}^2 = \left\| e_l + J_l \begin{bmatrix} \delta_{k+1}^{l+1/l} \\ \delta_k^{l+1/l} \end{bmatrix} \right\|_{\mathfrak{E}}^2 \quad (3.107)$$

où

$$J_l = \begin{bmatrix} \varphi_G \left(\delta_{k+1}^{l,k} \right) & 0 \\ 0 & \varphi_G \left(\delta_k^{l,k} \right) \\ \varphi_G \left(\delta_{k+1}^{l,k} \right) & 0 \\ \varphi_G \left(\delta_{k+1}^{l,T} \right) & 0 \end{bmatrix} = \begin{bmatrix} \mathcal{M} & 0 \\ 0 & \mathcal{S} \\ \mathcal{M} & 0 \\ \mathcal{L} & 0 \end{bmatrix} \quad (3.108)$$

avec $\mathcal{M} = \varphi_G \left(\delta_{k+1}^{l,k} \right)$, $\mathcal{S} = \varphi_G \left(\delta_k^{l,k} \right)$, $\mathcal{L} = \varphi_G \left(\delta_{k+1}^{l,T} \right)$ et

$$e_l = \begin{bmatrix} \delta_{k+1}^{l,k} \\ \delta_k^{l,k} \\ \delta_{k+1}^{l,k} \\ \delta_{k+1}^{l,T} \end{bmatrix} \quad (3.109)$$

La solution de (3.107) s'écrit alors :

$$\begin{bmatrix} \delta_{k+1}^{l+1/l} \\ \delta_k^{l+1/l} \end{bmatrix} = - \left(J_l^T \mathfrak{E}^{-1} J_l \right)^{-1} J_l^T \mathfrak{E}^{-1} e_l \quad (3.110)$$

où

$$\mathfrak{E}^{-1} = \begin{bmatrix} R_k^{-1} & -R_k^{-1} F_k & & \\ -F_k^T R_k^{-1} & F_k^T R_k^{-1} F_k + P_{k|k}^{-1} & & \\ & & -P_{k+1|k}^{-1} & \\ & & & P_{k+1|T}^{-1} \end{bmatrix} \quad (3.111)$$

Nous allons maintenant chercher à expliciter le calcul de (3.110). Pour cela, nous allons commencer par calculer le terme $\left(J_l^T \mathfrak{E}^{-1} J_l \right)^{-1}$ ainsi que le terme $J_l^T \mathfrak{E}^{-1} e_l$. Nous montrerons ensuite comment calculer l'incrément $\delta_{k+1}^{l+1/l}$ puis l'incrément $\delta_k^{l+1/l}$.

3.6.4.1 Calcul de $\left(J_l^T \mathfrak{E}^{-1} J_l \right)^{-1}$

Calculons l'inverse de chaque bloc de $\left(J_l^T \mathfrak{E}^{-1} J_l \right)^{-1}$:

$$\left(J_l^T \mathfrak{E}^{-1} J_l \right)^{-1} = \begin{bmatrix} \mathcal{M}^T \left(R_k^{-1} - P_{k+1|k}^{-1} \right) \mathcal{M} + \mathcal{L}^T P_{k+1|T}^{-1} \mathcal{L} & -\mathcal{M}^T R_k^{-1} F_k \mathcal{S} \\ -\mathcal{S}^T F_k^T R_k^{-1} \mathcal{M} & \mathcal{S}^T \left(F_k^T R_k^{-1} F_k + P_{k|k}^{-1} \right) \mathcal{S} \end{bmatrix}^{-1} \quad (3.112)$$

Il est possible de montrer que (voir annexe G.6) :

$$\begin{aligned} \left(J_l^T \mathfrak{E}^{-1} J_l \right)^{-1} = & \\ & \begin{bmatrix} \mathcal{L}^{-1} P_{k+1|T} \mathcal{L}^{-T} & \mathcal{L}^{-1} P_{k+1|T} \mathcal{L}^{-T} \mathcal{M}^T L_k^T \mathcal{S}^{-T} \\ \mathcal{S}^{-1} L_k \mathcal{M} \mathcal{L}^{-1} P_{k+1|T} \mathcal{L}^{-T} & \mathcal{S}^{-1} \left(P_{k|k} + L_k \left(\mathcal{M} \mathcal{L}^{-1} P_{k+1|T} \mathcal{L}^{-T} \mathcal{M}^T - P_{k+1|k} \right) L_k^T \right) \mathcal{S}^{-T} \end{bmatrix} \end{aligned} \quad (3.113)$$

où

$$L_k = P_{k|k} F_k^T P_{k+1|k}^{-1} \quad (3.114)$$

3.6.4.2 Calcul de $J_l^T \mathfrak{E}^{-1} e_l$

D'après (3.108) et (3.111) :

$$\begin{aligned}
 J_l^T \mathfrak{E}^{-1} e_l &= \begin{bmatrix} \mathcal{M}^T R_k^{-1} & -\mathcal{M}^T R_k^{-1} F_k & -\mathcal{M}^T P_{k+1|k}^{-1} & \mathcal{L}^T P_{k+1|T}^{-1} \\ -\mathcal{S}^T F_k^T R_k^{-1} & \mathcal{S}^T (F_k^T R_k^{-1} F_k + P_{k|k}^{-1}) & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_{k+1}^{l,k} \\ \delta_k^{l,k} \\ \delta_{k+1}^{l,T} \\ \delta_{k+1}^{l,T} \end{bmatrix} \\
 &= \begin{bmatrix} \mathcal{M}^T (R_k^{-1} - P_{k+1|k}^{-1}) \delta_{k+1}^{l,k} - \mathcal{M}^T R_k^{-1} F_k \delta_k^{l,k} + \mathcal{L}^T P_{k+1|T}^{-1} \delta_{k+1}^{l,T} \\ -\mathcal{S}^T F_k^T R_k^{-1} \delta_{k+1}^{l,k} + \mathcal{S}^T (F_k^T R_k^{-1} F_k + P_{k|k}^{-1}) \delta_k^{l,k} \end{bmatrix} \quad (3.115)
 \end{aligned}$$

3.6.4.3 Calcul de $\delta_{k+1}^{l+1/l}$

Supposons que $X_{k+1}^l = \mu_{k+1|T}$, alors d'après (3.103) $\delta_{k+1}^{l,T} = 0$ et $\mathcal{L} = Id$.

En utilisant (3.110), (3.113) et (3.115), nous obtenons alors :

$$\begin{aligned}
 \delta_{k+1}^{l+1/l} &= -P_{k+1|T} \left(\mathcal{M}^T (R_k^{-1} - P_{k+1|k}^{-1}) \delta_{k+1}^{l,k} - \mathcal{M}^T R_k^{-1} F_k \delta_k^{l,k} \right) \\
 &\quad - P_{k+1|T} \mathcal{M}^T L_k^T \mathcal{S}^{-T} \mathcal{S}^T \left(-F_k^T R_k^{-1} \delta_{k+1}^{l,k} + (F_k^T R_k^{-1} F_k + P_{k|k}^{-1}) \delta_k^{l,k} \right) \\
 &= -P_{k+1|T} \left(\mathcal{M}^T (R_k^{-1} - P_{k+1|k}^{-1}) \delta_{k+1}^{l,k} - \mathcal{M}^T R_k^{-1} F_k \delta_k^{l,k} \right) \\
 &\quad - P_{k+1|T} \mathcal{M}^T L_k^T \left(-F_k^T R_k^{-1} \delta_{k+1}^{l,k} + (F_k^T R_k^{-1} F_k + P_{k|k}^{-1}) \delta_k^{l,k} \right) \quad (3.116)
 \end{aligned}$$

En réunissant les termes communs, nous avons :

$$\begin{aligned}
 \delta_{k+1}^{l+1/l} &= -P_{k+1|T} \mathcal{M}^T \left((R_k^{-1} - P_{k+1|k}^{-1}) - L_k^T F_k^T R_k^{-1} \right) \delta_{k+1}^{l,k} \\
 &\quad + (-R_k^{-1} F_k + L_k^T (F_k^T R_k^{-1} F_k + P_{k|k}^{-1})) \delta_k^{l,k} \quad (3.117)
 \end{aligned}$$

D'après (G.12) le terme en $\delta_{k+1}^{l,k}$ est nul. Et d'après (G.16) le terme en $\delta_k^{l,k}$ est nul aussi. Donc en initialisant $X_{k+1}^0 = \mu_{k+1|T}$, nous avons montré que :

$$\delta_{k+1}^{l+1/l} = 0 \quad (3.118)$$

pour tout l . Par conséquent : $\delta_{k+1}^{l,T} = 0$ et $\mathcal{L} = Id$.

3.6.4.4 Calcul de $\delta_k^{l+1/l}$

En utilisant (3.110), (3.113) et (3.115), et le fait que $\delta_{k+1}^{l,T} = 0$ et $\mathcal{L} = Id$, nous obtenons :

$$\begin{aligned} \delta_k^{l+1/l} &= -\mathcal{S}^{-1} \left(L_k \mathcal{M} P_{k+1|T} \mathcal{M}^T \left(R_k^{-1} - (R_k + F_k P_{k|k} F_k^T)^{-1} \right) \right. \\ &\quad \left. - (P_{k|k} + L_k (\mathcal{M} P_{k+1|T} \mathcal{M}^T - P_{k+1|k}) L_k^T) F_k^T R_k^{-1} \right) \delta_{k+1}^{l,k} \\ &\quad - \mathcal{S}^{-1} \left(-L_k \mathcal{M} P_{k+1|T} \mathcal{M}^T R_k^{-1} F_k \right. \\ &\quad \left. + (P_{k|k} + L_k (\mathcal{M} P_{k+1|T} \mathcal{M}^T - P_{k+1|k}) L_k^T) (F_k^T R_k^{-1} F_k + P_{k|k}^{-1}) \right) \delta_k^{l,k} \end{aligned}$$

Il est possible de montrer que (voir annexe G.7) :

$$\delta_k^{l+1/l} = \mathcal{S}^{-1} L_k \delta_{k+1}^{l,k} \quad (3.120)$$

3.6.5 Algorithme LG-RTS

En initialisant $X_k^0 = \mu_{k|k}$ et $X_{k+1}^0 = \mu_{k+1|T}$, nous avons obtenu (3.118) et (3.120) :

$$\begin{aligned} \begin{bmatrix} \delta_{k+1}^{l+1/l} \\ \delta_k^{l+1/l} \end{bmatrix} &= - \left(J_l^T \mathfrak{E}^{-1} J_l \right)^{-1} J_l^T \mathfrak{E}^{-1} e^l \\ &= \begin{bmatrix} 0 \\ \mathcal{S}^{-1} (L_k \delta_{k+1}^{l,k} - \delta_k^{l,k}) \end{bmatrix} \end{aligned} \quad (3.121)$$

Or par définition :

$$\delta_k^{l+1/l} = \log_G^\vee \left(X_k^{l+1} (X_k^l)^{-1} \right) \quad (3.122)$$

et

$$\delta_{k+1}^{l,k} = \log_G^\vee \left(X_{k+1}^l \mu_{k+1|k}^{-1} \right) = \log_G^\vee \left(\mu_{k+1|T} \mu_{k+1|k}^{-1} \right) \quad (3.123)$$

donc nous avons :

$$\begin{aligned} \delta_k^{l+1,k} &= \log_G^\vee \left(X_k^{l+1} \mu_{k|k}^{-1} \right) \\ &= \log_G^\vee \left(X_k^{l+1} (X_k^l)^{-1} X_k^l \mu_{k|k}^{-1} \right) \\ &= \log_G^\vee \left(\exp_G^\wedge \left(\delta_k^{l+1/l} \right) \exp_G^\wedge \left(\delta_k^{l,k} \right) \right) \\ &\simeq \delta_k^{l,k} + \varphi_G \left(\delta_k^{l,k} \right) \delta_k^{l+1/l} \\ &= \delta_k^{l,k} + \mathcal{S} \delta_k^{l+1/l} \\ &= \delta_k^{l,k} + \mathcal{S} \left(-\mathcal{S}^{-1} (L_k \delta_{k+1}^{l,k} - \delta_k^{l,k}) \right) \\ &= L_k \delta_{k+1}^{l,k} \end{aligned} \quad (3.124)$$

Ce qui nous donne

$$\begin{aligned} X_k^{l+1} &= \exp_G^\wedge \left(L_k \delta_{k+1}^{l,k} \right) \mu_{k|k} \\ &= \exp_G^\wedge \left(L_k \log_G^\vee \left(\mu_{k+1|T} \mu_{k+1|k}^{-1} \right) \right) \mu_{k|k} \end{aligned} \quad (3.125)$$

La solution est obtenue après une seule itération et nous avons donc :

$$\hat{X}_k = \exp_G^\wedge \left(L_k \log_G^\vee \left(\mu_{k+1|T} \mu_{k+1|k}^{-1} \right) \right) \mu_{k|k} \quad (3.126)$$

On pose alors $\mu_{k|T} = \hat{X}_k$. Concernant la covariance, une approximation de Gauss-Laplace suivie d'une marginalisation de X_{k+1} nous indique que $P_{k|T}$ peut être approché par le bloc inférieur droit de (3.113) :

$$P_{k|T} = D' = \mathcal{S}^{-1} \left(P_{k|k} + L_k \left(\mathcal{M} P_{k+1|T} \mathcal{M}^T - P_{k+1|k} \right) L_k^T \right) \mathcal{S}^{-T} \quad (3.127)$$

L'algorithme présenté dans la partie précédente est appelé lisseur de Rauch-Tung-Striebel sur groupe de Lie (LG-RTS).

L'algorithme LG-RTS est résumé ci-après :

Algorithme 3.4 LG-RTS

Entrée : $\mu_{k|k}$, $P_{k|k}$, $\mu_{k+1|k}$, $P_{k+1|k}$, $\mu_{k+1|T}$, $P_{k+1|T}$

Sortie : $\mu_{k|T}$, $P_{k|T}$

$$\begin{aligned} L_k &= P_{k|k} F_k^T P_{k+1|k}^{-1} \\ \mu_{k|T} &= \exp_G^\wedge \left(L_k \log_G^\vee \left(\mu_{k+1|T} \mu_{k+1|k}^{-1} \right) \right) \mu_{k|k} \\ P_{k|T} &= \mathcal{S}^{-1} \left(P_{k|k} + L_k \left(\mathcal{M} P_{k+1|T} \mathcal{M}^T - P_{k+1|k} \right) L_k^T \right) \mathcal{S}^{-T} \end{aligned}$$

3.6.5.1 Une généralisation du RTS

Dans le cas spécifique où G , le groupe de Lie sur lequel évolue l'état X est un espace euclidien, alors l'algorithme LG-RTS se réduit au RTS. Cela vient du fait que le modèle proposé (voir paragraphe 3.6.2) se réduit alors au modèle du RTS (voir paragraphe 3.6.1) puisque la distribution normale concentrée sur groupe de Lie devient une distribution normale multivariée.

Les mêmes remarques que celles effectuées concernant les différences entre la mise à jour du LG-IEKF et celle de l'IEKF peuvent être faites.

En effet, dans l'équation du lissage de la covariance (3.127), les matrices $\mathcal{M} = \varphi_G \left(\log_G^\vee \left(\mu_{k+1|T} \mu_{k+1|k}^{-1} \right) \right)$, $\mathcal{S}^{-1} = \Phi_G \left(\log_G^\vee \left(\mu_{k|T} \mu_{k|k}^{-1} \right) \right)$ effectuent des reparamétrisations de la covariance (voir paragraphe 1.2.2). Détaillons ces reparamétrisations.

La matrice $P_{k+1|T}$ a un *sens* vis à vis de $\mu_{k+1|T}$ alors que $P_{k+1|k}$ a un *sens* vis à vis de $\mu_{k+1|k}$. Ainsi, pour pouvoir soustraire $P_{k+1|k}$ à $P_{k+1|T}$, le LG-RTS transforme d'abord $P_{k+1|T}$ avec la matrice \mathcal{M} afin que $P_{k+1|T}$ ait également un *sens* vis à vis de $\mu_{k+1|k}$.

La covariance finale $P_{k|T}$ doit avoir un *sens* vis à vis de $\mu_{k|T}$, or le terme $P_{k|k} + L_k (\mathcal{M}P_{k+1|T}\mathcal{M}^T - P_{k+1|k}) L_k^T$ a un *sens* par rapport à $\mu_{k|k}$. Par conséquent, ce terme est reparamétré par la matrice \mathcal{S}^{-1} afin qu'il ait un *sens* (voir paragraphe 1.2.2.2) non plus vis à vis de $\mu_{k|k}$ mais de la moyenne lissée $\mu_{k|T}$.

Remarquons finalement que, comme le LG-IEKF, au lieu d'un incrément additif (“+”), le LG-RTS utilise un incrément faisant intervenir “ \exp_G^\wedge ”. De la même manière, au lieu de considérer une erreur euclidienne (“−”), le LG-RTS emploie l'application “ \log_G^\vee ”.

En ce sens, le LG-RTS peut donc être vu comme une généralisation du RTS.

3.6.6 Application du LG-RTS sur un problème de lissage de la trajectoire d'une caméra

Nous avons choisi d'évaluer les performances du formalisme proposé par rapport à celle du LG-EKF sur un problème d'estimation de la trajectoire d'une caméra. L'objectif est d'estimer la pose $C_k \in SE(3)$ de la caméra en considérant un modèle de propagation où l'accélération est un bruit blanc Gaussien :

$$C_{k+1} = \exp_{SE(3)}^\wedge(v_k \Delta t) C_k \quad (3.128)$$

Par conséquent, la vitesse de la caméra $v \in \mathbb{R}^6$ est également estimée. Elle est décrite par le modèle suivant :

$$v_{k+1} = v_k + n_k \quad (3.129)$$

où n_k est un bruit blanc Gaussien centré de covariance R_{v_k} .

L'état X_k que nous estimons évolue donc sur le groupe de Lie $G = SE(3) \times \mathbb{R}^6$. Nous supposons que C_k est directement observé mais uniquement disponible un pas de temps sur 5 :

$$Z_k = \exp_{SE(3)}^\wedge(w_k) C_k \quad (3.130)$$

où w_k est un bruit blanc Gaussien centré de covariance Q_k .

Les algorithmes LG-EKF et LG-RTS appliqués à ces modèles sont présentés en annexe (voir Alg.H.2, Alg.H.3 et Alg.H.6).

Dans nos simulations, $\Delta t = 0.1$, $R_{v_k} = \text{diag}(0.1, 0.1, 0.1, 0.1, 0.1, 0.1)$ et $Q_k = \text{diag}(10^{-6}, 10^{-6}, 10^{-6}, 10^{-3}, 10^{-3}, 10^{-3})$.

	LG-EKF	LG-RTS
EQM (norm.unit)	55.7	30.5

Tableau 3.1 – Erreur Quadratique Moyenne sur la pose (position et orientation) de la caméra : $\frac{1}{N} \sum_{i=1}^N \left\| \log_{SE(3)}^V \left(C_k^{estimé} (C_k^{vrai})^{-1} \right) \right\|^2$ (100 trajectoires)

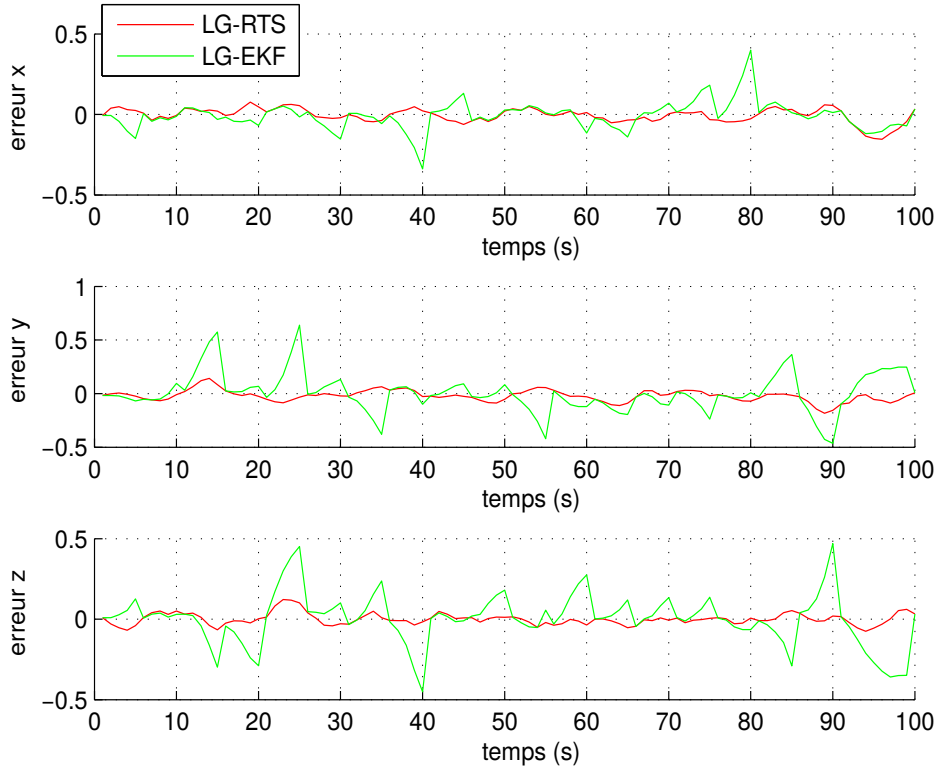


Figure 3.7 – Erreur d'estimation de la position de la caméra sur une trajectoire

La figure 3.7 représente l'erreur d'estimation de la position de la caméra pour une trajectoire. Le tableau 3.1 représente l'erreur quadratique moyenne de la pose de la caméra (position et orientation). Comme nous pouvons nous y attendre, pour cette application, le fait de prendre en compte l'information du futur, comme le fait le LG-RTS, permet d'améliorer sensiblement les performances du LG-EKF.

Cette combinaison des deux algorithmes, à savoir appliquer le LG-EKF pour filtrer la trajectoire puis employer le LG-RTS pour lisser la trajectoire filtrée, sera employée dans le chapitre suivant au sein d'un lisseur Rao-Blackwellisé destiné à estimer la trajectoire d'une caméra sur données réelles.

3.7 Résumé des contributions et conclusion

Dans ce chapitre, nous avons établi un lien entre l'algorithme de Gauss-Newton intrinsèque et le filtrage/lissage de Kalman sur groupe de Lie via l'approximation de Gauss-Laplace. En exploitant ce lien, nous avons montré comment obtenir une généralisation du filtre Kalman étendu itéré aux groupes de Lie. Nous avons désigné ce nouvel algorithme LG-IEKF. Nous avons également proposé une généralisation du lisseur de Rauch-Tung-Striebel aux groupes de Lie appelée LG-RTS.

Les performances du LG-IEKF ont été illustrées sur des problèmes de débruitage de transformations relatives à la fois sur données synthétiques mais également sur données réelles. Finalement, l'intérêt du LG-RTS a été démontré sur le problème du lissage de la trajectoire d'une caméra où la prise en compte de l'information future a permis d'améliorer sensiblement la sortie du LG-IEKF.

Cartographie et localisation à partir d'une caméra monoculaire

4.1	Introduction	133
4.2	Cartographie à partir d'une caméra monoculaire	134
4.2.1	Introduction	134
4.2.2	État de l'art	136
4.2.3	Architecture de VSLAM monoculaire proposée	138
4.2.4	Sélection d'images clés	138
4.2.5	Reconstruction d'une sous-carte	139
4.2.6	Estimation de la similitude 3D entre deux sous-cartes	142
4.2.7	Débruitage de similitudes 3D relatives	142
4.2.8	Limites de la méthode proposée	151
4.2.9	Résultats	152
4.3	Localisation à partir d'une caméra monoculaire dans un environnement cartographié	159
4.3.1	Introduction	159
4.3.2	État de l'art	163
4.3.3	Architecture du système de localisation en environnement intérieur	166
4.3.4	Un problème de suivi de cible sur $SE(3)$	168
4.3.5	Limites de la méthode proposée	176
4.3.6	Résultats	178
4.4	Résumé des contributions et conclusion	181

4.1 Introduction

Être capable d'estimer la position et l'orientation d'une caméra dans un environnement préalablement cartographié à partir de l'information contenue

dans son flux vidéo est devenu un problème central dans diverses applications, allant des services basés sur la localisation, aux véhicules autonomes, en passant par la reconnaissance d’activités [González D 2013].

Afin de répondre à ces besoins, dans ce chapitre, nous proposons tout d’abord une méthode permettant de cartographier automatiquement l’environnement à partir d’une vidéo d’apprentissage. Dans une seconde partie, nous décrivons une technique de localisation, reposant sur la carte de l’environnement ainsi obtenue, capable d’estimer la trajectoire d’une caméra uniquement à partir de son flux vidéo.

Que ce soit dans le contexte de la cartographie à partir d’une caméra monoculaire ou bien dans le contexte de la localisation de cette caméra monoculaire, nous verrons que nous serons amenés à estimer des paramètres évoluant sur des groupes de Lie. Nous mettrons donc à profit les concepts et algorithmes présentés dans les chapitres précédents.

4.2 Cartographie à partir d’une caméra monoculaire

4.2.1 Introduction

Cette partie est consacrée à la cartographie d’un environnement à partir d’une vidéo d’apprentissage issue d’une caméra monoculaire. Avant d’aller plus loin, il convient de préciser ce que nous entendons par “cartographie”. Dans notre contexte, ce terme signifie : estimation conjointe de la trajectoire de la caméra et de l’environnement dans lequel elle évolue (VSLAM en anglais pour Visual Simultaneous Localization And Mapping). Cette approche conjointe est indépendante de l’objectif final (reconstruire l’environnement ou estimer la trajectoire de la caméra). En effet, si on souhaite reconstruire l’environnement à partir des images de la vidéo, alors les poses des caméras doivent être préalablement estimées. Si au contraire l’objectif est d’obtenir la trajectoire de la caméra dans l’environnement, alors ce dernier doit être connu. Dans les deux cas, une approche d’estimation conjointe est nécessaire.

Ce problème du VSLAM est central en vision par ordinateur. En effet, de nombreuses applications, telles que la localisation d’une caméra [Sattler 2011 ; Li 2012] ou encore la réalité augmentée, supposent qu’un modèle de l’environnement a été préalablement reconstruit. Par conséquent, être capable de fournir un modèle précis de l’environnement est une condition sine qua non au bon fonctionnement de ces applications.

Des algorithmes de VSLAM, dédiés aux caméras stéréoscopiques, à la fois robustes, précis et capables de reconstruire de larges environnements ont été

proposés il y a quelques années [Konolige 2008 ; Mei 2009]. Cependant, ce type de caméra est encore peu répandu par rapport aux caméras monoculaires qui sont notamment présentes sur les smartphones. Ainsi, dans ce sous-chapitre, nous nous intéressons au problème du VSLAM monoculaire.

La principale difficulté du VSLAM monoculaire vis à vis du VSLAM stéréoscopique réside dans le fait que l'échelle de la scène n'est pas observée. Illustrons dès à présent cette difficulté. Prenons le cas d'une caméra monoculaire évoluant dans un environnement urbain. En visionnant la vidéo issue de cette caméra, il n'est pas possible de savoir si la caméra évolue dans une maquette d'une ville où les bâtiments font quelques centimètres de hauteur, ou bien s'il s'agit d'une ville réelle auquel cas les bâtiments font plusieurs mètres d'envergure.

N'ayant aucune information supplémentaire nous informant sur l'échelle de la scène, il est courant que, lors de la reconstruction de l'environnement, l'échelle d'une partie de la scène soit différente de l'échelle d'une autre partie. On parle alors de *dérive de l'échelle de la scène*. Par exemple, si une caméra filme une rue, où toutes les maisons sont identiques, en commençant au numéro 1 et en terminant au numéro 100, alors les maisons reconstruites numéro 1 et numéro 100 peuvent avoir des tailles totalement différentes. Aucune information ne contraint explicitement les reconstructions des maisons 1 et 100 à avoir la même envergure.

Une manière d'ajouter une information permettant de réduire cette dérive de l'échelle consiste à détecter *les fermetures de boucles*, c'est à dire les moments où la caméra revient dans un endroit qu'elle a déjà visité. Le fait de fermer les boucles permet ainsi d'ajouter l'information selon laquelle différents instants de la vidéo observent la même scène et ont donc la même échelle, réduisant par la même occasion la dérive de l'échelle entre ces instants.

Cependant, un grand environnement présente en général des endroits qui se ressemblent fortement (comme des bureaux par exemple, ou encore des couloirs). Dans ce contexte, de fausses fermetures de boucles peuvent être calculées (indiquant par exemple que la caméra a visité deux fois le même bureau alors qu'il s'agit de deux bureaux, certes similaires, mais situés à deux endroits différents dans l'environnement) conduisant à une carte de l'environnement erronée.

Dans cette partie, nous proposons un algorithme de VSLAM monoculaire à la fois robuste et capable de cartographier de longues séquences vidéo, tout en fermant les boucles.

Afin de cartographier l'environnement, notre méthode reconstruit tout d'abord des sous-cartes, c'est à dire des morceaux de la trajectoire de la caméra et de l'environnement, à l'aide d'une méthode basée sur ledit Problème des Orientations Connues [Olsson 2010]. L'idée de construire notre algorithme de

reconstruction autour du Problème des Orientations Connues provient du fait que, dans le domaine de la reconstruction à partir d'images non ordonnées [Moulon 2013], ce formalisme a conduit à l'élaboration des méthodes fournissant actuellement les résultats de l'état de l'art. La qualité des résultats obtenus est principalement due au fait que la reconstruction de la scène s'effectue alors de manière globale et non pas de manière incrémentale (voir [Moulon 2013]) réduisant ainsi la dérive de l'échelle de la scène à l'intérieur de chaque sous-carte.

Dans le but d'obtenir une reconstruction complète de l'environnement, il est nécessaire d'aligner les sous-cartes tout en prenant garde à ne pas prendre en compte les fermetures de boucles aberrantes. Pour ce faire, nous construisons un graphe de similitudes 3D calculées entre les sous-cartes qui représentent les fermetures de boucles. Afin de détecter et supprimer les fermetures de boucles aberrantes, nous proposons un algorithme à la fois simple et efficace reposant sur un test statistique sur le groupe de Lie des similitudes $Sim(3)$. Finalement, dans le but d'obtenir une méthode capable de reconstruire de larges environnements, nous développons une approche de type Propagation de Croyance (LBP en anglais pour Loopy Belief Propagation) capable d'aligner un nombre important de sous-cartes efficacement.

Ce découpage en deux grandes étapes consistant à reconstruire des sous-cartes puis à les aligner va permettre, comme nous allons le voir, d'obtenir une solution au problème du VSLAM monoculaire, à la fois calculatoirement efficace et capable d'opérer dans des conditions difficiles où les algorithmes de l'état de l'art rencontrent des difficultés.

4.2.2 État de l'art

Le problème du VSLAM monoculaire a été étudié depuis plus de vingt ans. Ainsi, un état de l'art exhaustif est difficilement réalisable. Ici, nous proposons plutôt une description des méthodes les plus récentes et nous mettons l'accent sur leurs différences vis à vis de la méthode que nous proposons. Le lecteur pourra se référer à [Eade 2008 ; Strasdat 2012] pour un état de l'art plus large sur le sujet.

La grande majorité des approches récentes, ainsi que celle que nous proposons, sont constituées de deux principaux modules.

1. Un module d'odométrie visuelle (VO en anglais pour Visual Odometry) qui permet d'estimer les poses de caméra ainsi que le modèle 3D à partir de plusieurs images consécutives de la vidéo. Dans [Clemente 2007] et [Eade 2008 ; Eade 2007], le module de VO construit des sous-cartes en utilisant respectivement un filtre de Kalman et un algorithme d'ajustement

de faisceaux¹ [Engels 2005] (BA en anglais pour Bundle Adjustment) incrémental. [Strasdat 2010a] ne construit pas explicitement de sous-cartes mais emploie un BA incrémental avec une fenêtre glissante comprenant les dix dernières images clés de la vidéo. Finalement, dans [Engel 2014], une approche semi-dense est utilisée pour estimer la carte de profondeur de chaque image clé. Par rapport aux approches de l'état de l'art, nous proposons un module de VO différent qui se base sur le Problème des Orientations Connues [Olsson 2010]. Cela nous permet d'estimer chaque sous-carte globalement (c'est à dire de manière non incrémentale) et ainsi d'éviter une dérive de l'échelle à l'intérieur de la sous-carte tout en détectant et supprimant les données aberrantes grâce à un programme linéaire (LP en anglais pour Linear Programming).

2. Un module de fermeture de boucles qui permet de réduire la dérive de l'échelle. Pour ce faire, les fermetures de boucles entre les sous-cartes, comme dans [Clemente 2007; Eade 2008] par exemple, ou bien directement entre les images clés comme dans [Strasdat 2010a; Engel 2014], sont tout d'abord détectées. Puis, afin de prendre en compte ces fermetures de boucles pour réduire la dérive de l'échelle, une fonction coût, dont les paramètres estimés diffèrent selon les approches, est généralement minimisée. Pour cela, [Clemente 2007] utilise un algorithme hiérarchique proposé dans [Estrada 2005], [Eade 2008] emploie une descente de gradient pré-conditionnée alors que [Strasdat 2010a] et [Engel 2014] se servent d'un algorithme de Levenberg-Marquardt. Il est important de remarquer qu'aucune des approches précédemment citées n'est capable de gérer des fermetures de boucles aberrantes, ce qui accroît fortement leurs chances d'obtenir une reconstruction erronée, surtout dans un environnement relativement grand. Par rapport aux approches de l'état de l'art, nous proposons un algorithme de propagation de croyance capable de gérer un grand nombre de fermetures de boucles sans la nécessité d'une initialisation. De plus, nous proposons un algorithme simple et efficace de détection de fermeture de boucles aberrantes.

Il convient de noter que la méthode proposée dans [Lim 2014] est proche de celle décrite dans [Strasdat 2010a]. Cependant, plusieurs modifications ont été apportées et ont permis d'obtenir les résultats de l'état de l'art sur la base de données KITTI. Ainsi, dans la suite nous comparons notre méthode à celles de [Lim 2014] et [Engel 2014].

1. Un algorithme d'ajustement de faisceaux consiste, à partir de correspondances entre des points d'intérêt présents dans plusieurs images, à estimer conjointement les poses des caméras ayant capturé ces images et les coordonnées 3D des points d'intérêt.

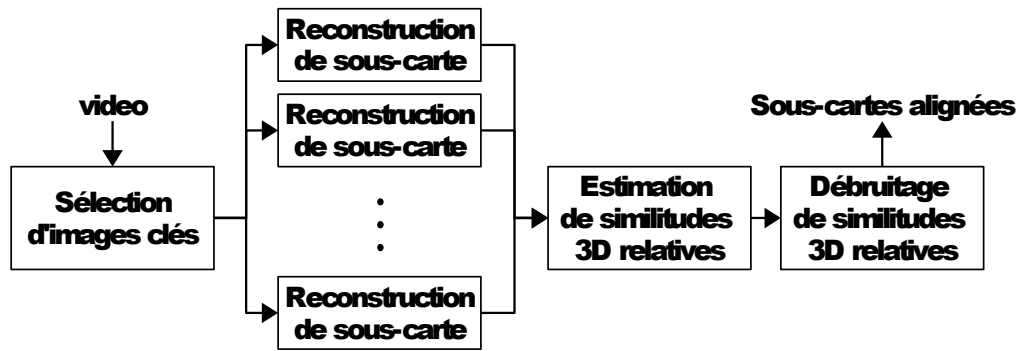


Figure 4.1 – Architecture de VSLAM monoculaire proposée

4.2.3 Architecture de VSLAM monoculaire proposée

L'architecture de notre méthode de VLSAM monoculaire comporte 4 modules :

- sélection d'images clés
- reconstruction de sous-carte
- estimation de similitudes 3D relatives entre des paires de sous-cartes (détection des fermetures de boucles)
- débruitage de similitudes 3D relatives (alignement des sous-cartes)

qui sont reliés entre eux comme présenté en figure 4.1. Nous présentons maintenant chaque module individuellement.

Remarquons que nous nous plaçons dans le cas où la caméra a été préalablement calibrée et les images rectifiées de manière à ce que le modèle de caméra sténopé puisse s'appliquer. Ainsi, nous travaillons dans le plan focal et non pas dans le plan image. Le lecteur qui n'est pas familier avec ces concepts peut consulter [Hartley 2000].

4.2.4 Sélection d'images clés

Sélectionner des images clés parmi toutes les images de la vidéo est nécessaire pour obtenir un temps de calcul raisonnable. Afin de réaliser cette sélection, nous détectons et suivons des points d'intérêt (PoI en anglais pour Point of Interest) de Harris dans la vidéo à l'aide d'un algorithme de suivi de type Lucas-Kanade [Lucas 1981].

La première image de la vidéo est une image clé par défaut. L'algorithme suit ensuite les PoI dans les images suivantes, jusqu'à ce que la distance euclidienne moyenne entre les PoI de l'image courante et ceux de l'image clé précédente soit supérieure à un seuil (typiquement 5% de la largeur de l'image). Cela nous permet donc de sélectionner peu d'images clés lorsque la caméra bouge lentement et au contraire beaucoup d'images clés lorsqu'elle bouge vite.

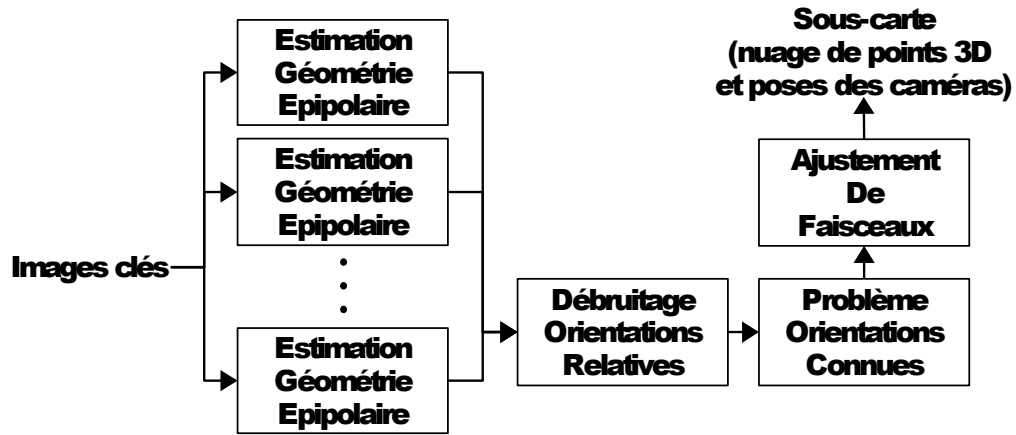


Figure 4.2 – Architecture proposée pour la reconstruction d'une sous-carte

De plus, l'algorithme fonctionne pour tout type de mouvement de la caméra (rotation pure, translation pure, etc.).

4.2.5 Reconstruction d'une sous-carte

Après avoir sélectionné les images clés, nous définissons des groupes de L images clés consécutives ayant un recouvrement d'un facteur $\frac{1}{2}$. Cela signifie qu'un groupe partage ses $\frac{L}{2}$ premières images clés avec le groupe précédent et ses $\frac{L}{2}$ dernières images clés avec le groupe suivant.

Pour chaque groupe indépendamment, nous appliquons un algorithme de reconstruction basé sur le Problème des Orientations Connues [Olsson 2010] pour obtenir un ensemble de sous-cartes. Le Problème des Orientations Connues stipule que si l'on dispose des coordonnées de points d'intérêt vus dans plusieurs images ainsi que les orientations exactes de ces images, il est alors possible de retrouver les positions 3D des points d'intérêt ainsi que les positions 3D des images en utilisant un programme linéaire. De plus, l'algorithme est capable, dans une certaine mesure, de détecter et supprimer les coordonnées aberrantes de points d'intérêts.

Une étape cruciale consiste donc à fournir les meilleurs estimés possibles des orientations afin d'obtenir la reconstruction la plus fidèle possible en sortie du Problème des Orientations Connues.

Remarquons que notre méthode de reconstruction de sous-carte est similaire à celle proposée dans [Olsson 2011] à cela près que dans notre cas, les images que nous considérons sont temporellement consécutives, ce qui constitue un avantage dont nous allons tirer parti. Une illustration graphique de notre procédure de reconstruction de sous-carte est présentée en figure 4.2. Nous décrivons maintenant notre méthode pour une sous-carte.

4.2.5.1 Estimation de la géométrie épipolaire

Tout d'abord, les points d'intérêt SURF [Bay 2006] sont extraits des images clés. Ensuite les descripteurs de ces points SURF sont appariés entre des paires d'images clés (nous ne considérons pas toutes les paires possibles, mais uniquement un sous-ensemble, dans le but de conserver un coût calculatoire raisonnable). Cela permet de compléter l'information apportée par les points de Harris (qui ont été suivis dans le module de Sélection d'images clés) et de "fermer les boucles" à l'intérieur de chaque sous-carte.

Ensuite, pour chaque paire d'images clés considérées, les poses des deux caméras ainsi que le nuage de points 3D sont estimés en couplant un RANSAC avec un algorithme des 5 points [Nister 2004] suivi d'une triangulation des points 3D. Le résultat obtenu est utilisé comme initialisation d'un BA, minimisant l'erreur de reprojection des points 3D dans les deux images clés (qui peut être minimisée à l'aide d'un LG-GN² par exemple), suivi d'une approximation de Gauss-Laplace intrinsèque afin d'avoir une estimation de l'incertitude des paramètres estimés.

4.2.5.2 Débruitage des orientations relatives

Nous pouvons maintenant récupérer les orientations relatives ainsi que leurs covariances calculées à l'étape précédente afin d'estimer le plus précisément possible les orientations absolues. Cette étape, que nous avons appelée débruitage d'orientations relatives dans le chapitre précédent, peut être résolue de différentes manières. Dans la suite de ce chapitre, nous présentons une solution au problème du débruitage de similitudes 3D relatives (Alg.4.1 et Alg.4.2) qui peut être directement appliquée pour des orientations (car une orientation 3D est une similitude 3D avec une translation nulle et un facteur d'échelle de 1). Par conséquent, nous sommes capables d'estimer précisément les orientations 3D tout en détectant les potentielles orientations relatives aberrantes.

4.2.5.3 Reconstruction 3D

Les orientations absolues étant estimées, nous employons un LP pour résoudre le Problème des Orientations Connues, ce qui nous donne la position 3D de chaque pose de caméra ainsi que les positions des points 3D. Cette étape, quant à elle, est rendue robuste aux correspondances aberrantes entre images en utilisant le LP³ proposé dans [Olsson 2010].

2. A chaque itération du LG-GN, afin de résoudre efficacement le système linéaire, il est possible de tirer parti de la structure du problème et d'utiliser le complément de Schur [Konolige 2010].

3. Nous utilisons la librairie d'optimisation MOSEK pour résoudre chaque LP.

Finalement, un BA est appliqué, utilisant comme initialisation la sortie du LP et les orientations absolues placées en entrée du LP, suivi d'une approximation de Gauss-Laplace intrinsèque. De cette manière, la distribution a posteriori de chaque sous-carte est approchée par une distribution normale concentrée sur groupe de Lie.

Remarques concernant l'approximation de Gauss-Laplace intrinsèque appliquée au BA Dans le contexte du BA, le problème que nous souhaitons résoudre comporte 7 degrés de liberté (ddl) singuliers. En effet, appliquer une similitude 3D à la fois au nuage de points 3D et aux poses des caméras ne modifie pas la reprojection des points 3D dans les images. Par conséquent, lorsqu'on souhaite appliquer une approximation de Gauss-Laplace intrinsèque, la pseudo-Hessienne, qui est de la forme $J^T \Sigma^{-1} J$, est dégénérée de 7 degrés de liberté.

Une façon d'obtenir la covariance des paramètres dans ce cas [Morris 2001] consiste à calculer la pseudo-inverse de la matrice $J^T \Sigma^{-1} J$ en utilisant une décomposition en valeurs singulières, où les 7 plus petites valeurs propres sont laissées à 0.

Remarquons cependant que cette solution aboutit à une matrice de covariance singulière.

Dans un contexte euclidien, cela ne poserait pas de problème, car il est possible de définir une distribution normale multivariée avec une matrice de covariance singulière en considérant la mesure de Lebesgue restreinte au sous-espace non dégénéré.

En revanche, dans le cas d'un groupe de Lie, le fait de retirer des degrés de liberté n'engendre pas en général un nouveau groupe de Lie. Par conséquent, l'existence d'une mesure de Haar invariante à gauche ou à droite, sur laquelle est fondée la définition de la distribution normale concentrée, n'est dans ce cas plus garantie.

D'un point de vue pratique, nous ne calculons jamais la covariance complète mais seulement les covariances marginales des paramètres. Les points 3D ayant des directions dégénérées sont supprimés. Et les covariances marginales des poses des caméras sont en pratique toujours de rang plein. Cette technique nous permet d'obtenir un algorithme numériquement très stable, capable de quantifier l'incertitude des paramètres estimés par le BA.

Cette méthode que nous avons proposée est capable de fournir une reconstruction précise d'une sous-carte en étant robuste à chaque étape aux potentielles données aberrantes. Elle fonctionne même dans des cas difficiles où la caméra effectue une faible translation ainsi que lorsque l'environnement n'est pas parfaitement statique (voir partie 4.3.6).

4.2.6 Estimation de la similitude 3D entre deux sous-cartes

Après avoir reconstruit chaque sous-carte indépendamment, nous souhaitons les aligner pour avoir à la fois un nuage de points 3D de tout l'environnement ainsi que la trajectoire complète de la caméra. Une manière d'aligner les sous-cartes est de commencer par calculer les similitudes 3D entre des paires de sous-cartes, ce qui constitue l'objectif de cette partie, et d'ensuite chercher à estimer les similitudes 3D absolues, ce qui constitue l'objectif de la partie suivante.

Une technique de type sac de mots [Sivic 2009] est tout d'abord appliquée aux descripteurs SURF des points 3D de toutes les sous-cartes afin d'obtenir un unique descripteur pour chaque sous-carte. Ensuite, pour chaque sous-carte, nous essayons de calculer une similitude 3D entre cette sous-carte et ses 10 plus proches voisines. Ici plus proche voisine s'entend au sens de la somme des valeurs absolues entre chaque composante des deux descripteurs. Ainsi, au lieu d'essayer de calculer des similitudes entre toutes les paires de sous-cartes, cette technique nous permet de réduire significativement le temps de calcul.

La similitude 3D entre deux sous-cartes i et j est obtenue de la manière suivante :

1. Les descripteurs SURF des points 3D (ces descripteurs ont été obtenus en calculant la moyenne des descripteurs SURF des points 2D) des deux sous-cartes sont appariés en utilisant un arbre k-d.
2. Un algorithme des 3 points [Umeyama 1991] combiné à un RANSAC est ensuite appliqué pour estimer la similitude 3D. Cette similitude est raffinée en minimisant la somme des erreurs euclidiennes pondérée par la covariance des points 3D (estimée à la fin de la reconstruction de chaque sous-carte) grâce à un LG-GN. Finalement, une distribution normale concentrée sur $Sim(3)$ est ajustée en utilisant une approximation de Gauss-Laplace intrinsèque. Ainsi, en sortie de cette étape, nous disposons d'une similitude 3D $Z_{ij} \in Sim(3)$ entre le référentiel de la sous-carte j et le référentiel de la sous-carte i , de même que la covariance Σ_{ij}^i de cet estimé.

4.2.7 Débruitage de similitudes 3D relatives

Étant donné les similitudes 3D relatives estimées comme décrit dans la partie précédente, nous souhaitons maintenant estimer les similitudes 3D absolues qui vont nous permettre d'aligner toutes les sous-cartes dans un référentiel commun.

4.2.7.1 Sans données aberrantes

En supposant que les similitudes 3D relatives dont nous disposons ne contiennent pas de données aberrantes, c'est à dire pas de fausses fermetures de boucles, le problème que nous souhaitons résoudre consiste à minimiser le critère suivant :

$$\operatorname{argmin}_{\{X_{iS}\}_{i \in \mathcal{V}}} \left(\sum_{(i,j) \in \mathcal{E}} \left\| \log_{Sim(3)}^{\vee} (Z_{ij} X_{jS} X_{iS}^{-1}) \right\|_{\Sigma_{ij}^i}^2 \right) \quad (4.1)$$

où $Z_{ij} \in Sim(3)$ est une similitude 3D relative bruitée entre le référentiel de la sous-carte j et le référentiel de la sous-carte i . S est le référentiel absolu (ou référentiel global) alors que les X_{iS} et les X_{jS} sont les similitudes 3D absolues que nous souhaitons estimer. Les matrices de covariance Σ_{ij}^i correspondent aux matrices obtenues en appliquant l'approximation de Gauss-Laplace intrinsèque comme expliqué dans la partie 4.2.6. Le critère (4.1) que nous cherchons à minimiser provient du modèle génératif :

$$Z_{ij} = \exp_{Sim(3)}^{\wedge} (b_{ij}^i) X_{iR} X_{jR}^{-1} \quad (4.2)$$

où $b_{ij}^i \sim \mathcal{N}_{\mathbb{R}^7} (b_{ij}^i; \mathbf{0}_{7 \times 1}, \Sigma_{ij}^i)$ est un bruit blanc gaussien.

Remarquons que ce problème peut être vu comme un problème d'inférence dans un graphe de facteurs $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, où chaque nœud \mathcal{V}_i correspond à une similitude globale X_{iS} et chaque lien \mathcal{E}_{ij} représente une similitude relative Z_{ij} reliant les deux nœuds \mathcal{V}_i et \mathcal{V}_j . Un exemple de graphe est donné en figure 4.5a.

4.2.7.2 En présence de données aberrantes

Le problème (4.1) utilise une norme L2 qui est “sensible” aux fermetures de boucles aberrantes. Or, dans notre contexte, puisque la caméra évolue dans un grand environnement, il est très fréquent d'observer des scènes visuellement similaires, voire même dupliquées, ce qui conduit à un nombre important de fermetures de boucles aberrantes (voir Fig.4.3). Ainsi, avant de chercher à résoudre le problème (4.1), il est intéressant de chercher à détecter et supprimer les fermetures de boucles aberrantes.

4.2.7.3 État de l'art

Dans la partie 3.5.7, nous avons proposé une méthode, basée sur le LG-IEKF muni d'un fenêtrage statistique, permettant de résoudre le problème (4.1). Cependant, cette méthode est limitée à un nombre restreint de similitudes 3D absolues N (typiquement $N < 500$) en raison de la taille de la matrice de covariance l'état ($7N \times 7N$).

Dans [Zach 2010], une approche collectant les résidus de boucles calculés dans le graphe est proposée, afin d'identifier les fermetures aberrantes. Cependant, le fait de calculer les résidus de toutes les boucles est trop coûteux d'un point de vue calculatoire et la taille maximale des boucles considérées est donc limitée à 6.

Finalement, différentes méthodes venant de la communauté robotique ont été proposées [Sunderhauf 2012 ; Latif 2013 ; Olson 2012]. Ces approches emploient un algorithme de Levenberg-Marquardt pour simultanément identifier les liens aberrants dans le graphe et estimer les paramètres souhaités. En revanche, ces algorithmes n'ont pas été employés pour estimer des similitudes 3D.

Contrairement à ces approches, nous montrons qu'en exploitant la structure du problème traité et notre connaissance de la covariance des liens (rendue possible grâce au formalisme de distribution normale concentrée), il est possible de séparer les étapes de détection des fermetures aberrantes et d'estimation des similitudes globales en deux étapes distinctes. Ainsi, dans la partie suivante, nous présentons un algorithme à la fois simple et efficace, permettant de détecter les fermetures de boucles aberrantes. Nous décrivons ensuite un nouvel algorithme de propagation de croyance, permettant d'estimer un grand nombre de similitudes 3D efficacement.

4.2.7.4 Algorithme de détection de fermetures de boucles aberrantes

Afin de détecter les fermetures de boucles aberrantes, nous faisons l'hypothèse que les similitudes 3D relatives $Z_{(i-1)i}$ calculées entre deux sous-cartes temporellement consécutives, c'est à dire les sous-cartes ayant des images clés en commun, ne contiennent pas de données aberrantes. Cette hypothèse, classique en SLAM, est en pratique vérifiée dans toutes nos expériences (voir partie 4.2.9). Remarquons tout de même que cette hypothèse n'est plus respectée, si par exemple, la main du cameraman vient cacher le champ de vision. Cependant, dans le contexte où la scène observée est quasi-statique, cette hypothèse semble vérifiée.

Dans un graphe sans données aberrantes, intégrer les similitudes relatives le long d'un cycle produit un "petit" résidu, c'est à dire :

$$\epsilon^T P^{-1} \epsilon < t_{\chi^2} \quad (4.3)$$

où ϵ est le résidu du cycle, P est sa covariance et t_{χ^2} est un seuil dont la valeur est basée sur une valeur-p du χ^2 (7) [Fisher 1949]. Le résidu et la covariance d'un cycle peuvent être calculés efficacement en utilisant les deux équations (4.4) et (4.5) qui permettent d'intégrer/composer deux similitudes relatives et

Algorithme 4.1 Algorithme de détection de fermetures de boucles aberrantes

Entrée : $\{Z_{ij}\}_{1 \leq i < j \leq N}$ (similitudes 3D relatives), $\{\Sigma_{ij}^i\}_{1 \leq i < j \leq N}$ (matrices de covariance), t_{χ^2} (valeur-p du χ^2)

Sortie : \mathcal{E} (ensemble de fermetures de boucles non aberrantes)

1. Initialiser un graphe vide $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$
 2. Ajouter le nœud X_{1S} à \mathcal{V}
 3. Pour k allant de 2 à N
 - (a) Ajouter X_{kS} à \mathcal{V}
 - (b) Ajouter $\{Z_{(k-1)k}, \Sigma_{(k-1)k}^{k-1}\}$ à \mathcal{E}
 - (c) Pour chaque fermeture de boucle Z_{lk} avec $l < k$
 - i. Trouver le chemin de plus court entre X_{kS} et X_{lS} dans \mathcal{G} (à l'aide de l'algorithme de Dijkstra par exemple)
 - ii. Calculer le résidu ϵ de la boucle et sa covariance P
 - iii. Si $\epsilon^T P^{-1} \epsilon < t_{\chi^2}$ alors ajouter $\{Z_{lk}, \Sigma_{lk}^l\}$ à \mathcal{E}
-

leurs covariances :

$$Z_{kl} = Z_{km}Z_{ml} \quad (4.4)$$

$$\Sigma_{kl}^k \simeq \Sigma_{km}^k + \text{Ad}_{\text{Sim}(3)}(Z_{km}) \Sigma_{ml}^m \text{Ad}_{\text{Sim}(3)}^T(Z_{km})^T \quad (4.5)$$

Un algorithme naïf de détection de données aberrantes consiste, pour une similitude relative Z_{kl} , à calculer la covariance et le résidu de la boucle formée par la composition des similitudes relatives temporellement consécutives, i.e $Z_{kl}Z_{l(l-1)}Z_{(l-1)(l-2)}\dots Z_{(k-2)(k-1)}Z_{(k-1)k}$, puis à appliquer le test (4.3). Toutefois, cet algorithme naïf ne fonctionne pas pour de longues vidéos (voir Fig.4.3c) car la longueur des boucles calculées ne cesse d'augmenter, tout comme l'incertitude des boucles qui finit par être si grande que le résidu de boucles contenant des données aberrantes passe sous le seuil du test (4.3).

Nous proposons maintenant un algorithme capable de détecter les données aberrantes pour de longues vidéos (voir Alg.4.1). L'algorithme vérifie les fermetures de boucles les unes après les autres, non plus en considérant le cycle impliquant les similitudes relatives temporellement consécutives, mais le cycle le plus court⁴ parmi toutes les similitudes relatives déjà vérifiées comme non aberrantes.

4. Le cycle le plus court s'entend ici au sens du nombre de liens, c'est à dire que chaque lien "coûte" 1. Nous utilisons l'algorithme de Dijkstra pour trouver le cycle le plus court entre deux nœuds.

De cette manière, la longueur des boucles considérées n'est plus liée à la durée de la vidéo, ce qui permet de bien mieux détecter les données aberrantes. Un exemple de résultat de l'Alg.4.1 est présenté en figure 4.3d.

4.2.7.5 Algorithme de propagation de croyance

Dans cette partie, nous proposons un algorithme de propagation de croyance (LS-RSA en anglais pour Large Scale Relative Similarity Averaging) qui exploite la structure du problème (4.1) pour estimer efficacement un grand nombre de similitudes absolues à partir de similitudes relatives classées comme non aberrantes par l'algorithme 4.1.

L'algorithme consiste à partitionner le graphe original \mathcal{G} en N_S sous-graphes $\{\mathcal{G}^k = \{\mathcal{V}^k, \mathcal{E}^k\}\}_{k=1:N_S}$ (voir Fig.4.5a). Notre approche fonctionne ensuite de la manière suivante :

- résoudre le problème suivant pour chaque sous-graphe \mathcal{G}^k indépendamment (voir Fig.4.5b) :

$$\begin{aligned} \underset{\{X_{iR_k}\}_{i \in \mathcal{V}^k}}{\operatorname{argmin}} \left(\sum_{(i,j) \in \mathcal{E}^k} \left\| \log_{Sim(3)}^{\vee} (Z_{ij} X_{jR_k} X_{iR_k}^{-1}) \right\|_{\Sigma_{ij}^i}^2 \right. \\ \left. + \sum_{i \in \mathcal{V}^k} \left\| \log_{Sim(3)}^{\vee} (Z_{iR_k} X_{iR_k}^{-1}) \right\|_{\Sigma_{iR_k}^i}^2 \right) \end{aligned} \quad (4.6)$$

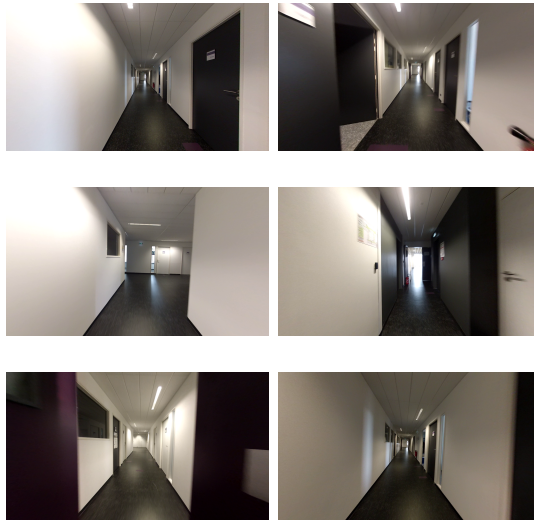
où chaque Z_{iR_k} est une similitude 3D ayant une covariance $\Sigma_{iR_k}^i$ qui peut être interprétée comme un message envoyé par les autres sous-graphes au nœud \mathcal{V}_i^k de \mathcal{G}^k ;

- calculer les messages, c'est à dire les similitudes 3D Z_{iR_k} et leurs covariances $\Sigma_{iR_k}^i$, en construisant et résolvant un super graphe $\mathcal{G}^{Super} = \{\mathcal{V}^{Super}, \mathcal{E}^{Super}\}$ (voir Fig.4.5c).

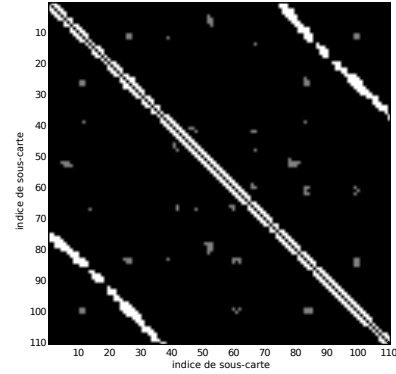
Chaque sous-graphe peut être optimisé en parallèle, ce qui rend ce nouvel algorithme efficace. Dans la suite de cette partie, nous ne différencions pas, pour simplifier la présentation, une variable d'optimisation X_{jR_k} de son estimé \hat{X}_{jR_k} .

Le pseudo-code du LS-RSA est présenté dans l'Alg.4.2. Nous décrivons maintenant chaque étape de cet algorithme en détail :

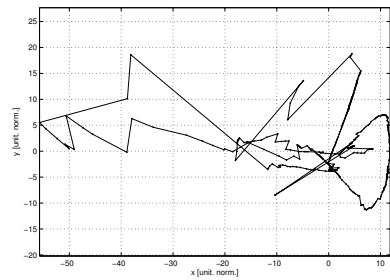
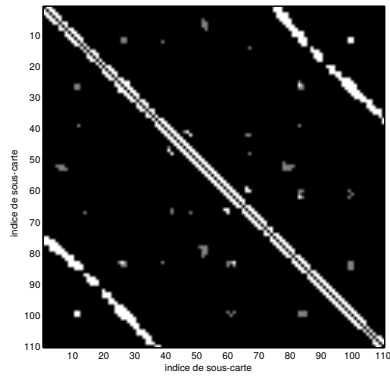
1. Partitionnement du graphe : La première étape consiste à partitionner temporellement le graphe original \mathcal{G} en N_S sous-graphes de taille maximale n où $n > 1$. Dans le reste des explications, nous supposons, pour ne pas compliquer inutilement l'exposé, que N est un multiple de N_S i.e chaque sous-graphe est constitué d'exactly n nœuds. Ici, le terme "temporellement" signifie que \mathcal{G} est partitionné en supprimant les



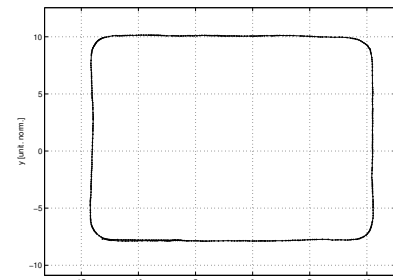
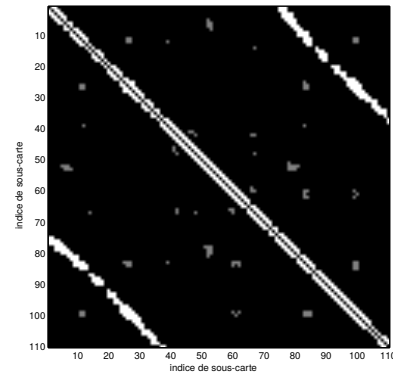
(a) Exemples d'images de la vidéo



(b) Vérité terrain

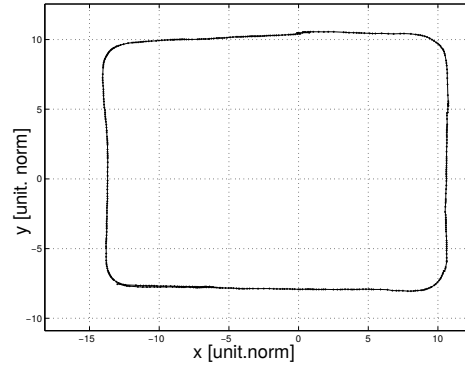


(c) (Haut) Détection de fermetures de boucles aberrantes en utilisant les cycles formés par les similitudes 3D temporellement consécutives. Plusieurs fermetures aberrantes sont détectées comme non aberrantes. (Bas) La trajectoire de la caméra n'est pas correctement estimée.

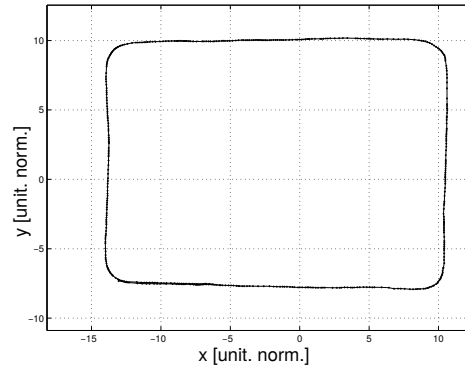


(d) (Haut) Résultat de l'Alg. 4.1. Les fermetures aberrantes sont parfaitement détectées. (Bas) La trajectoire de la caméra est correctement estimée (la trajectoire forme un rectangle presque parfait).

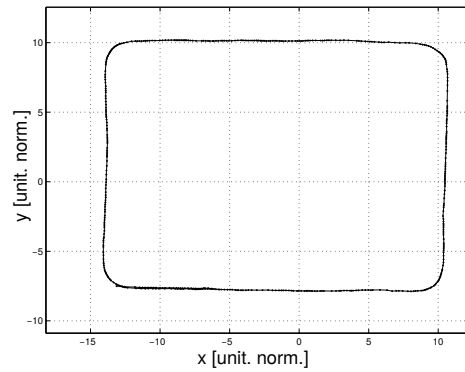
Figure 4.3 – Exemple de résultat sur une vidéo prise dans le couloir d'un bâtiment (le couloir forme un rectangle) ($t_{\chi^2} = 16$, $n = 10$). Dans une matrice de classification des données aberrantes, un pixel blanc indique une donnée non aberrante, un pixel gris indique une donnée aberrante et un pixel noir indique simplement qu'il n'y a pas d'observation.



(a) itération 1, résidu 24.46

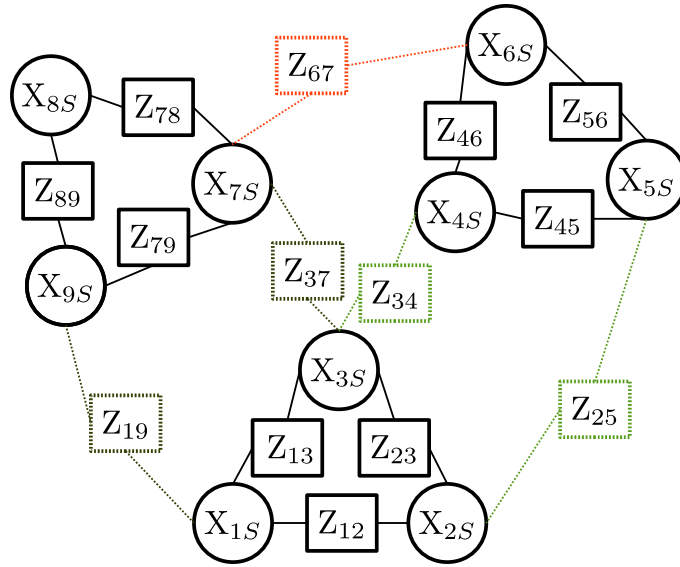


(b) itération 2, résidu 5.31

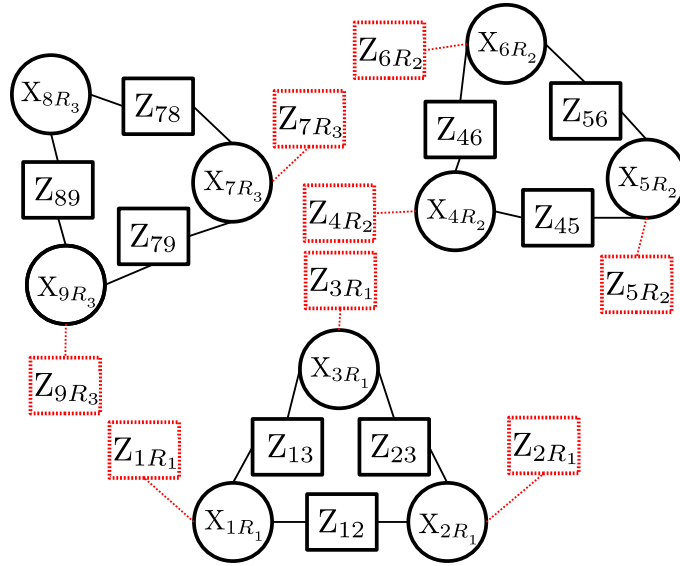


(c) itération 5, résidu 0.46

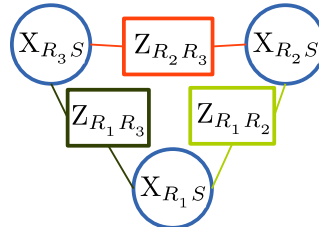
Figure 4.4 – Illustration des itérations de l'algorithme de propagation de croyance sur la séquence vidéo présentée en figure 4.3 ($n = 10$)



(a) Exemple d'un graphe de facteurs représentant le problème initial. Le fait de supprimer les facteurs en pointillés produit un graphe partitionné temporellement constitué de 3 sous-graphes de taille $n = 3$.



(b) Graphe de facteurs impliqué dans l'étape 3) de Alg. 4.2. Les sous-graphes sont déconnectés, par conséquent l'inférence peut être effectuée de manière indépendante pour chaque sous-graphe et en parallèle.



(c) Super graphe de facteurs impliqué dans les étapes 4) et 5) de Alg. 4.2.

Figure 4.5 – Illustrations des graphes de facteurs impliqués dans notre algorithme LS-RSA

facteurs qui connectent les ensembles suivants de nœuds : $\{X_{iS}\}_{i=1:n}$, $\{X_{iS}\}_{i=n+1:2n}$, ..., $\{X_{iS}\}_{i=N-n+1:N}$ (voir Fig.4.5a). Les facteurs supprimés sont appelés inter-facteurs.

2. Messages

- (a) Initialisation des messages : initialiser chaque message Z_{iR_k} avec la matrice identité et une matrice de covariance $\Sigma_{iR_k}^i$ avec une valeur infinie et aller à l'étape 3).
- (b) Calcul des messages : en utilisant les super similitudes absolues estimées à l'itération précédente (étape 5), nous pouvons calculer les messages qui vont être envoyés entre les nœuds des sous-graphes. Un nœud envoie un message à un autre nœud si les deux nœuds sont connectés par un inter-facteur. Pour chaque inter-facteur Z_{ij} , un message est créé en calculant la similitude Z_{iR_k} et sa covariance $\Sigma_{iR_k}^i$ de la manière suivante :

$$Z_{iR_k} = Z_{ij} X_{jR_l} X_{R_l S} X_{R_k S}^{-1} \quad (4.7)$$

$$\begin{aligned} \Sigma_{iR_k}^i &= \text{Ad}_{\text{Sim}(3)}(Z_{ij}) \left[P_{jR_l}^j + \text{Ad}_{\text{Sim}(3)}(X_{jR_l}) \left\{ P_{R_l S}^{R_l} \right. \right. \\ &\quad \left. \left. + \text{Ad}_{\text{Sim}(3)}(X_{R_l S} X_{R_k S}^{-1}) P_{R_k S}^{R_k} \text{Ad}_{\text{Sim}(3)}(X_{R_l S} X_{R_k S}^{-1})^T \right\} \text{Ad}_{\text{Sim}(3)}(X_{jR_l})^T \right] \\ &\quad \text{Ad}_{\text{Sim}(3)}(Z_{ij})^T + \Sigma_{ij}^i \end{aligned} \quad (4.8)$$

Cette étape peut être interprétée comme l'envoi d'un message de X_{jR_l} vers X_{iR_k} . Si un nœud X_{iR_k} reçoit plusieurs messages i.e plusieurs Z_{iR_k} ont été calculés puisque X_{iR_k} est connecté à plusieurs inter-facteurs, nous employons une moyenne de Karcher (voir [Barfoot 2014] paragraphe IV), c'est à dire un LG-GN suivi d'une approximation de Gauss-Laplace, pour résumer ces différents messages en un message unique.

3) Optimisation des sous-graphes : pour chaque sous-graphe \mathcal{G}^k , les similitudes absolues $\{X_{iR_k}\}_{i \in \mathcal{V}^k}$ ainsi que les covariances marginales de la distribution a posteriori $\{P_{iR_k}^i\}_{i \in \mathcal{V}^k}$ sont estimées en utilisant un LG-GN pour résoudre (4.6) suivi par une approximation de Gauss-Laplace. Remarquons que si cette étape ne contient qu'un seul sous-graphe ($N_S = 1$), alors l'algorithme s'arrête et retourne son résultat. Cette étape est illustrée par la figure 4.5b.

4) Construction du super-graphe : nous construisons maintenant un super-graphe $\mathcal{G}^{\text{Super}} = \{\mathcal{V}^{\text{Super}}, \mathcal{E}^{\text{Super}}\}$ (voir Fig.4.5c) à partir de la sortie de l'étape 3) et des inter-facteurs. Les liens de ce super-graphe sont des similitudes relatives entre les référentiels $\{R_k\}_{k=1:N_S}$ que nous appelons super-facteurs. Chaque

inter-facteur Z_{ij} de covariance Σ_{ij}^i produit le super-facteur suivant :

$$Z_{R_k R_l} = X_{i R_k}^{-1} Z_{ij} X_{j R_l} \quad (4.9)$$

de covariance :

$$\begin{aligned} \Sigma_{R_k R_l}^{R_k} = & \text{Ad}_{\text{Sim}(3)} \left(X_{i R_k}^{-1} \right) \left(P_{i R_k}^i + \Sigma_{ij}^i \right. \\ & \left. + \text{Ad}_{\text{Sim}(3)} (Z_{ij}) P_{j R_l}^j \text{Ad}_{\text{Sim}(3)} (Z_{ij})^T \right) \text{Ad}_{\text{Sim}(3)} \left(X_{i R_k}^{-1} \right)^T \end{aligned} \quad (4.10)$$

Puisque chaque inter-facteur engendre un super-facteur, il est possible de se retrouver avec plusieurs super-facteurs entre deux nœuds du super-graphe. Lorsque cela arrive, nous employons une moyenne de Karcher pour obtenir un unique super-facteur entre ces deux nœuds.

5) Optimisation du super-graphe : une fois que le super-graphe est construit, nous utilisons l'algorithme LS-RSA pour estimer les similitudes absolues $\{X_{R_k S}\}_{k=1:N_S}$ et leurs covariances $\{P_{R_k S}^{R_k}\}_{k=1:N_S}$. Remarquons que, pendant cette étape, le LS-RSA est appelé de manière récursive jusqu'à ce que l'étape 1) ne construise qu'un seul sous-graphe, ce qui conduit l'algorithme à se terminer à l'étape 3). Ainsi, le LS-RSA construit une série de super-graphes les uns au dessus des autres et peut donc être vu comme un algorithme hiérarchique.

6) Les similitudes absolues que nous souhaitons estimer, i.e $\{X_{i S}\}_{i \in \mathcal{V}}$ et $\{P_{i S}^i\}_{i \in \mathcal{V}}$ sont finalement obtenues de la manière suivante :

$$X_{i S} = X_{i R_k} X_{R_k S} \quad (4.11)$$

$$P_{i S}^i = P_{i R_k}^i + \text{Ad}_{\text{Sim}(3)} (X_{i R_k}) P_{R_k S}^{R_k} \text{Ad}_{\text{Sim}(3)} (X_{i R_k})^T \quad (4.12)$$

et le résidu de (4.1) peut également être calculé. Si ce résidu est plus grand que celui obtenu à l'itération précédente, alors l'algorithme s'arrête, sinon on retourne à l'étape 2.b).

Le résultat obtenu peut être utilisé comme initialisation d'un LG-GN global.

Dans la partie 4.2.9, nous démontrons sur plusieurs vidéos que le LS-RSA, qui est certes un algorithme heuristique, permet de fournir un résultat satisfaisant au problème du débruitage de similitudes relatives.

4.2.8 Limites de la méthode proposée

L'algorithme de VSLAM monoculaire que nous avons proposé présente plusieurs limites.

Tout d'abord, dans chaque sous-carte, le mouvement de la caméra ne doit pas être une pure rotation sinon les nuages de points 3D ne peuvent pas être estimés. Néanmoins, grâce au Problème des Orientations Connues, l'algorithme

Algorithme 4.2 Algorithme de propagation de croyance (LS-RSA)

Entrée : $\{Z_{ij}\}_{1 \leq i < j \leq N}$ (similitudes 3D relatives), $\{\Sigma_{ij}^i\}_{1 \leq i < j \leq N}$ (covariances), n (taille d'un sous-graphe)

Sortie : $\{X_{iS}\}_{1 \leq i \leq N}$ (similitudes 3D absolues), $\{P_{iS}^i\}_{1 \leq i \leq N}$ (covariances marginales)

- 1) Partitionner le graphe.
 - 2.a) Initialiser les messages et aller à 3).
 - 2.b) Calculer les nouveaux messages.
 - 3) Résoudre eq.(4.6) pour chaque sous-graphe.
 - 4) Construire le super-graphe.
 - 5) Appliquer le LS-RSA (appel récursif) au super-graphe.
 - 6) Calculer les similitudes absolues, i.e $\{X_{iS}\}_{i \in \mathcal{V}}$ et $\{P_{iS}^i\}_{i \in \mathcal{V}}$, ainsi que le résidu de eq.(4.1). Si le résidu n'a pas été réduit, sortir, sinon aller à 2.b).
-

proposé est capable de fournir de bons résultats pour de faibles translations de la caméra (voir Fig.4.11).

Deuxièmement, l'environnement 3D que nous souhaitons reconstruire doit être statique. Remarquons cependant que le fait que nous ayons utilisé des approches capables d'exclure les données aberrantes à chaque étage de la méthode permet à des objets, tels que des voitures dans la base de données KITTI, de bouger dans l'environnement (voir partie 4.2.9.2).

Troisièmement, si les similitudes 3D relatives temporellement consécutives contiennent des données aberrantes, alors l'Alg.4.1 ne permettra pas de détecter les données aberrantes. Toutefois, dans tous nos tests, ce cas n'est jamais survenu puisque les scènes reconstruites sont toutes quasi-statiques.

Finalement, la méthode proposée prend environ 5 heures pour traiter une vidéo de 20 minutes (codé en Matlab), en utilisant les paramètres précisés dans le paragraphe 4.2.9, ce qui ne permet pas d'obtenir un résultat en temps réel requis dans certaines applications.

4.2.9 Résultats

Nous comparons maintenant les performances de l'approche proposée aux algorithmes de l'état de l'art [Engel 2014] et [Lim 2014] sur les bases de données TUM et KITTI ainsi que sur d'autres séquences vidéos. Dans toutes ces expériences, les paramètres de notre approche ont été optimisés à la main sur une vidéo, une fois pour toutes ($L = 16$, $n = 10$, $t_{\chi^2} = 16$, ainsi que plusieurs autres paramètres tels que les seuils des RANSAC). Le fait d'augmenter L au-delà de 16 n'a pas conduit à de meilleurs résultats.

Concernant le problème du VSLAM monoculaire, le modèle 3D et la trajectoire de la caméra sont estimés à un facteur d'échelle près. Par conséquent, dans le reste de cette partie, lorsque nous comparons les résultats obtenus vis à vis de la vérité terrain, nous recalons d'abord les trajectoires en estimant une similitude 3D minimisant la distance entre la trajectoire estimée de la caméra et la vérité terrain à l'aide d'un LG-GN.

4.2.9.1 Résultats quantitatifs sur la base de données TUM

Dans [Engel 2014], la base de données TUM RGB-D [Sturm 2012] est utilisée pour évaluer leur algorithme. Par conséquent, nous avons choisi cette même base de données pour évaluer quantitativement les performances de notre approche. Dans la figure 4.6, le RMSE absolu de la trajectoire estimée (cm) [Sturm 2012] de notre approche, [Engel 2014], [Engel 2013], [Klein 2007], [Kerl 2013] et [Endres 2012] sont présentés⁵.

Pour chaque séquence vidéo, notre approche produit un RMSE plus petit que pour l'approche de [Engel 2014]. Les meilleures performances de notre approche sont probablement dues au fait qu'une sous-carte est estimée à partir de plusieurs images clés, alors que dans [Engel 2014], une fois qu'une image clé est sélectionnée, la carte de profondeur de l'image clé précédente n'est plus mise à jour.

Nous pouvons également observer que les résultats de notre approche tendent vers ceux des algorithmes de l'état de l'art en RGB-D SLAM [Kerl 2013] qui utilisent une caméra RGB-D au lieu d'une caméra RGB monoculaire. Dans la figure 4.7, la trajectoire estimée de la caméra avec notre approche sur la séquence FR2/desk est présentée.

4.2.9.2 Résultats qualitatifs sur la base de données KITTI

Dans [Lim 2014], la base de données KITTI [Geiger 2013] est utilisée pour évaluer leur algorithme. Par conséquent, nous avons choisi cette même base de données pour évaluer les performances de notre approche.

Dans la figure 4.8, les trajectoires estimées des caméras par notre approche et celle de [Lim 2014] sont comparées à la vérité terrain. Pour chaque trajectoire, la trajectoire estimée par notre méthode est plus proche de la vérité terrain que celle estimée par [Lim 2014]. Ce meilleur résultat est probablement dû au fait que, dans ces séquences, l'environnement n'est pas complètement statique (des voitures sont en mouvement). Ainsi, notre méthode est capable, grâce à

5. Les résultats de [Engel 2014; Engel 2013; Klein 2007; Kerl 2013; Endres 2012] sont reportés depuis la figure 9 de [Engel 2014].

	Methode Proposée [Engel 2014] [Engel 2013] [Klein 2007]				[Kerl 2013]	[Endres 2012]
Utilise une caméra 3D	Non	Non	Non	Non	Oui	Oui
fr2/desk	2.22	4.52	13.50	x	1.77	9.5
fr2/xyz	1.28	1.47	3.79	24.28	1.18	2.6

Figure 4.6 – Résultats sur la base de données TUM RGB-D. Les chiffres représentent le RMSE absolu de la trajectoire (cm) [Sturm 2012].

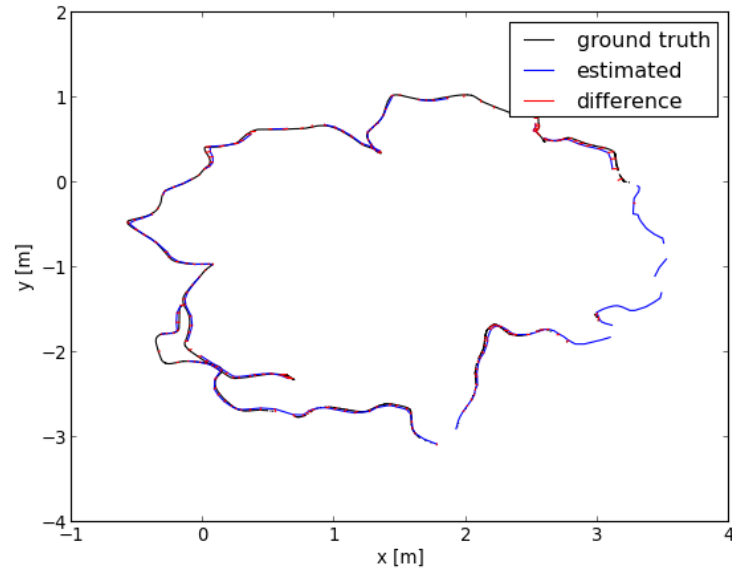


Figure 4.7 – Trajectoire de la caméra estimée avec la méthode proposée sur la séquence fr2/desk de la base de données TUM RGB-D. Cette courbe a été obtenue en utilisant l’outil web disponible sur la page web de la base de données TUM RGB-D.

l’exclusion des données aberrantes, de ne pas prendre en compte les objets non statiques dans ces vidéos.

Nous présentons également les résultats de notre approche sur les séquences 13 et 15 de la base de données KITTI (Fig.4.9 et Fig.4.10) pour lesquelles la vérité terrain n’est pas fournie. Nous disposons néanmoins d’une image de cette vérité terrain disponible sur le site internet KITTI. Dans la figure 4.9, nous montrons que notre approche obtient de meilleurs résultats que l’approche de [Lim 2014]. Le résultat de [Lim 2014] n’est pas disponible sur la séquence 15. A la place, nous comparons notre résultat à celui de l’algorithme de l’état de l’art utilisant un scanner laser Velodyne [Zhang 2015].

4.2.9.3 Résultats qualitatifs sur d’autres vidéos difficiles

Nous présentons ici des résultats sur des vidéos difficiles prises avec une caméra dite “rolling shutter”. Les vidéos contiennent du flou de mouvement, les environnements sont parfois peu texturés et les trajectoires contiennent de faibles translations de la caméra.

Dans la figure 4.3d, nous montrons la trajectoire de la caméra estimée dans le couloir d’un bâtiment (le couloir forme une rectangle autour de l’étage). Nous pouvons voir que la trajectoire estimée forme un rectangle plat presque parfait. Sur cette séquence, l’algorithme de suivi de [Engel 2014] a échoué.

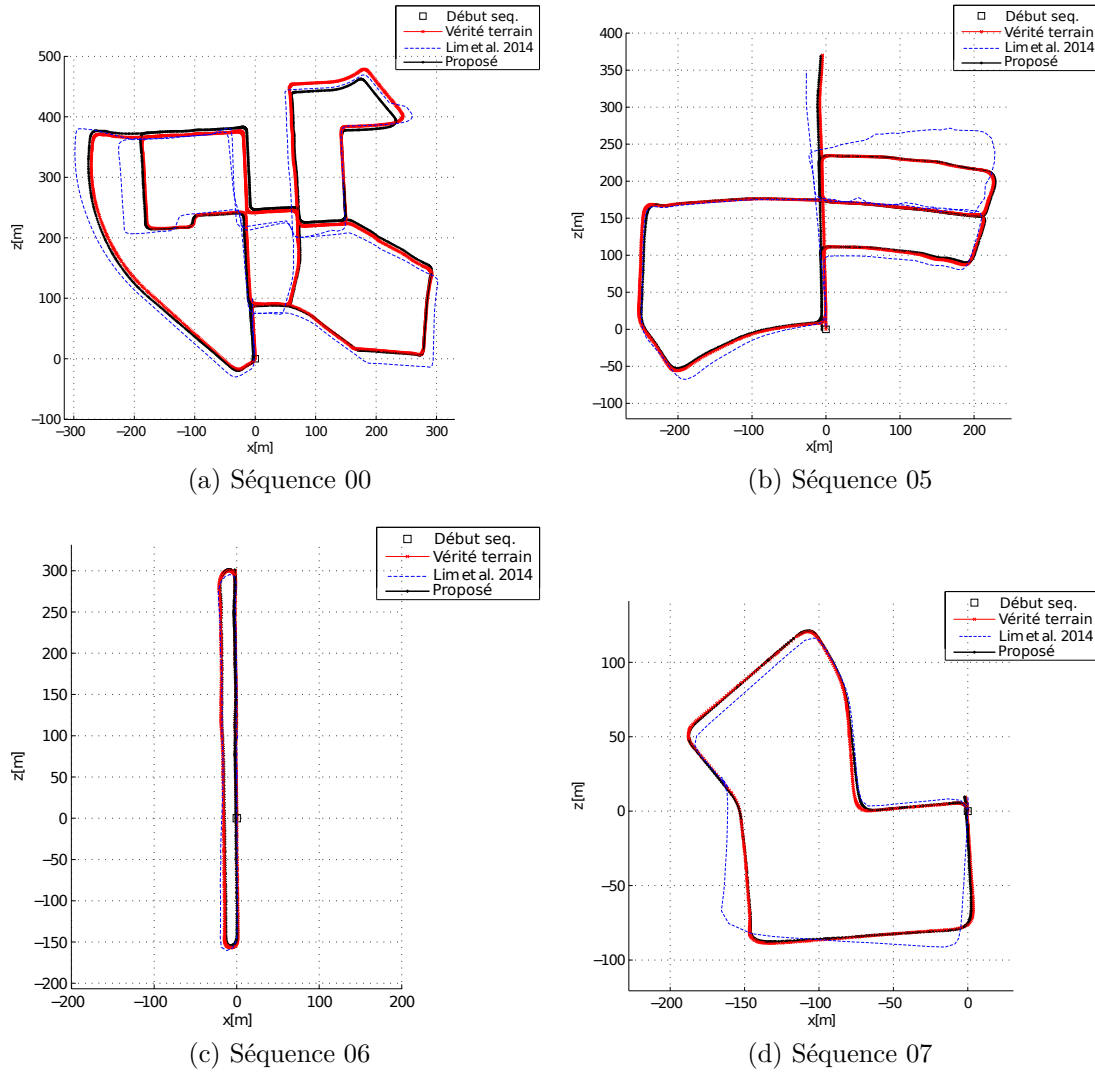
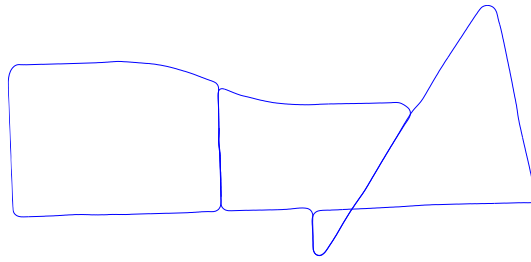
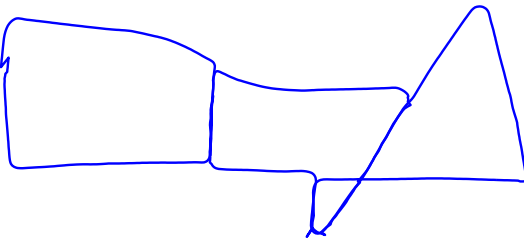


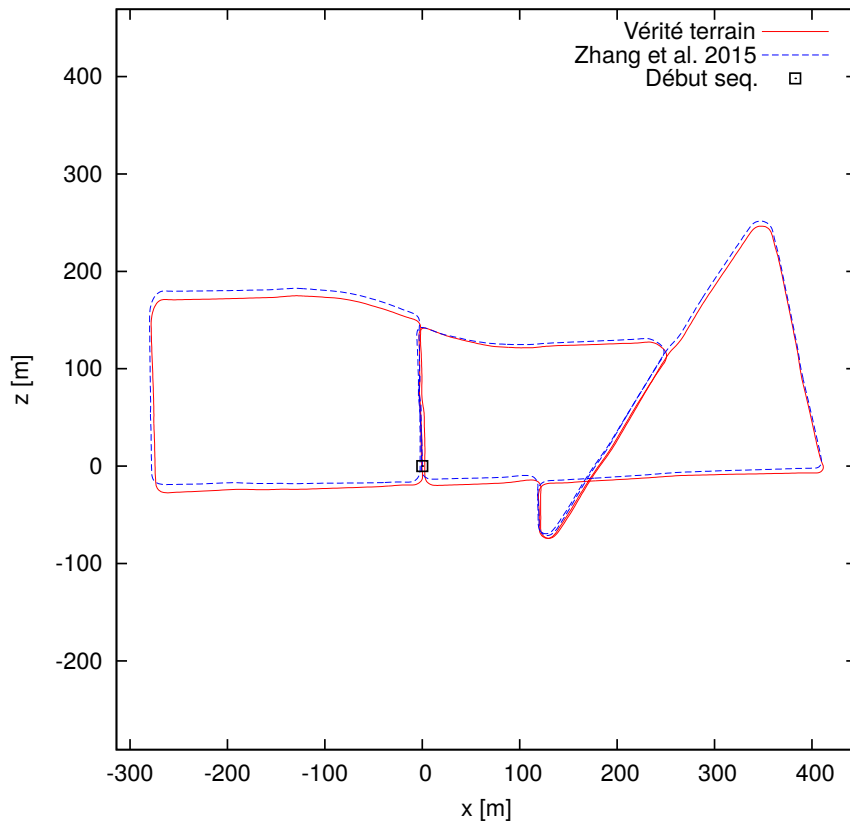
Figure 4.8 – Comparaison qualitative des trajectoires estimées avec l'approche proposée et celle de [Lim 2014] sur plusieurs séquences de la base de données KITTI. La plupart du temps, la trajectoire estimée par notre méthode se superpose avec la vérité terrain, contrairement à la trajectoire obtenue par [Lim 2014] qui dévie.



(a) Notre méthode (utilisant une caméra monoculaire)

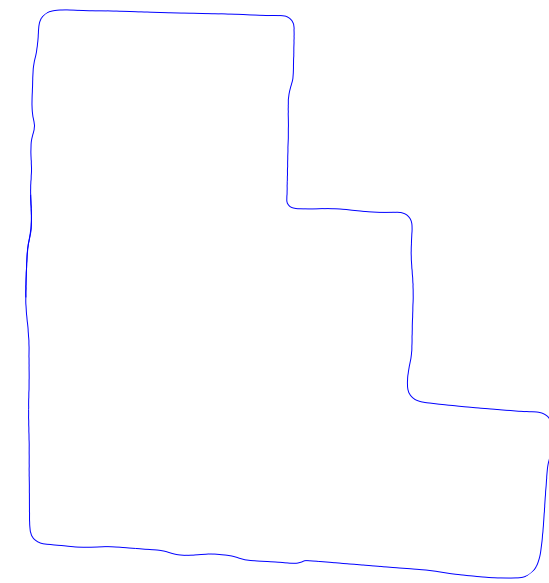


(b) Résultat de [Lim 2014] (utilisant une caméra monoculaire)

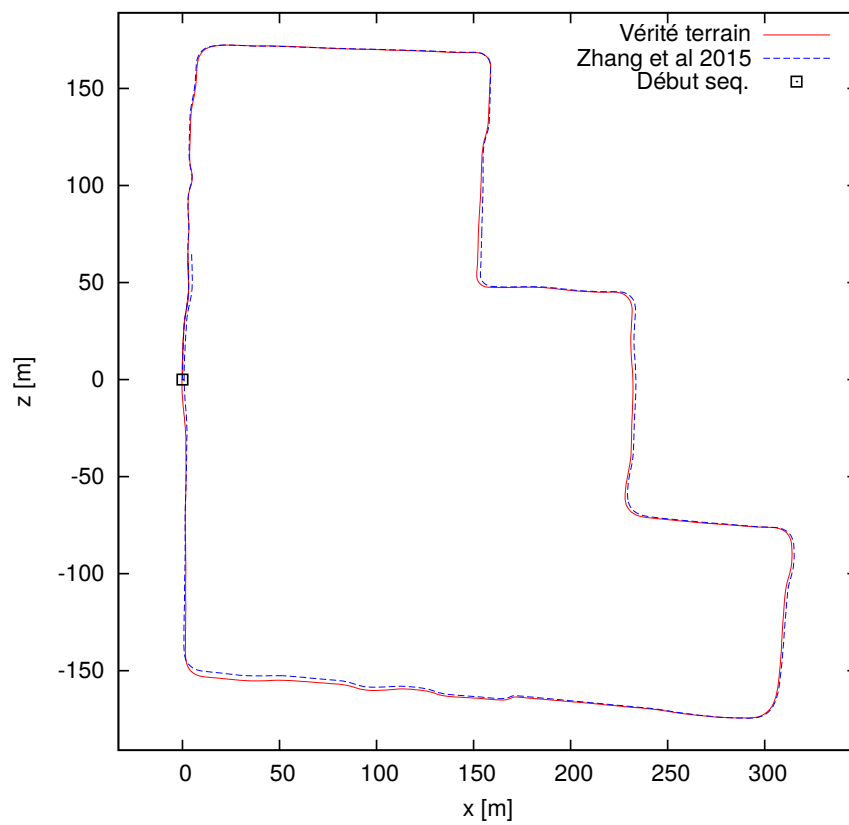


(c) Résultat de l'état de l'art [Zhang 2015] (utilisant un scanner laser Velodyne)

Figure 4.9 – Comparaison qualitative sur la séquence 13 de la base de données KITTI.



(a) Notre méthode (utilisant une caméra monoculaire)



(b) Résultat de l'état de l'art [Zhang 2015] (utilisant un scanner laser Velodyne)

Figure 4.10 – Comparaison qualitative sur la séquence 15 de la base de données KITTI.

Dans la figure 4.11, la trajectoire estimée d'une caméra montée sur une table tournante (la caméra est orientée vers l'extérieur) est présentée. Pour cette séquence, la translation de la caméra est faible, le rayon de la table étant de 10 cm. La trajectoire réelle de la caméra est un cercle parfait. Nous pouvons voir que la trajectoire produite par notre méthode est bien plus proche d'un cercle parfait que celle obtenue avec [Engel 2014].

Nous montrons finalement que notre approche est capable d'estimer la trajectoire d'une caméra évoluant dans un appartement. Ici la séquence vidéo contient 10000 images. Après avoir estimé la trajectoire de la caméra avec notre algorithme, nous l'alignons manuellement sur le plan de l'appartement (voir Fig. 4.12). Comme nous pouvons le voir, la trajectoire estimée est cohérente avec le plan de l'appartement, c'est à dire que la trajectoire visite les différentes pièces sans traverser les murs (la trajectoire emprunte bien les portes).

Un vidéo illustrant le système fonctionnant sur une vidéo de 20 minutes est disponible à l'adresse suivante :

<https://sites.google.com/site/guillaumbourmaud/home/videos>

4.3 Localisation à partir d'une caméra monoculaire dans un environnement cartographié

4.3.1 Introduction

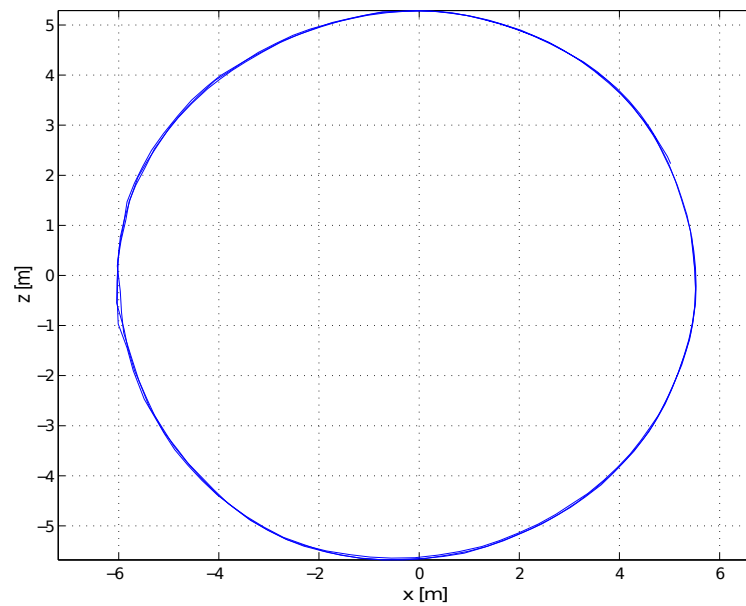
Par rapport à d'autres technologies telles que les systèmes RFID (Radio Fréquence Identification) ou les ondes WiFi, une caméra portée est un capteur passif à bas coût, ne nécessitant pas de modification de l'environnement dans lequel elle évolue. Par conséquent, être capable d'estimer précisément à la fois la position et l'orientation d'une caméra évoluant dans un environnement préalablement cartographié ouvre des perspectives applicatives lorsque l'environnement lui-même ne peut pas être modifié.

Dans cette partie, nous nous intéressons au problème de la localisation d'une caméra en environnement intérieur (VIL en anglais pour Visual Indoor Localization). Nous considérons le cas où l'environnement a été préalablement cartographié en appliquant l'algorithme de VSLAM monoculaire présenté dans le chapitre précédent à une vidéo d'apprentissage. Nous disposons donc notamment d'une "base d'images" (les images clés de la vidéo d'apprentissage) et de leurs "coordonnées" (les poses de ces images clés vivant sur le groupe de Lie $SE(3)$).

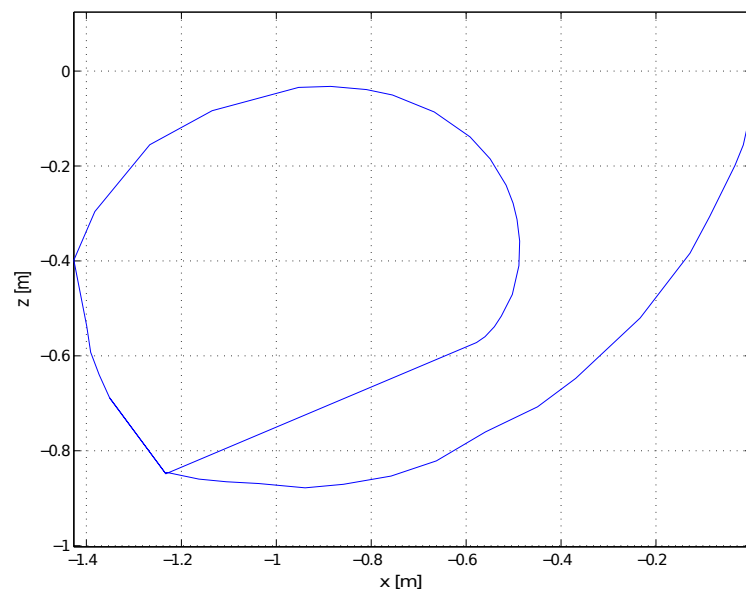
Dans notre contexte, la caméra est portée par une personne effectuant des



(a) Exemples d'images de la vidéo



(b) Notre approche



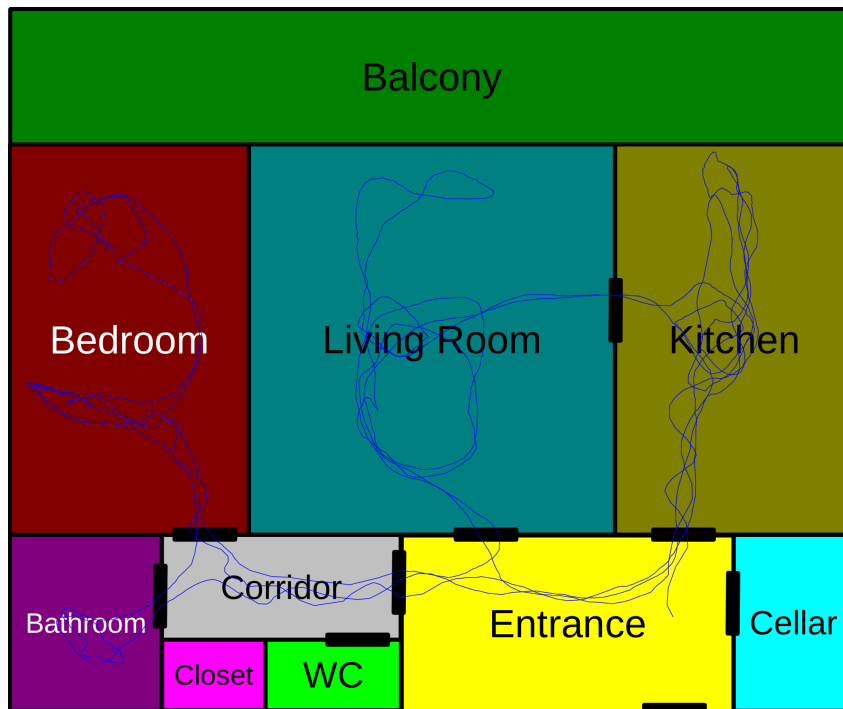
(c) Résultat de [Engel 2014]

Figure 4.11 – Résultat sur la séquence d'un cercle parfait

4.3. Localisation à partir d'une caméra monoculaire dans un environnement cartographié



(a) Exemples d'images de la vidéo



(b) Trajectoire estimée de la caméra (ligne bleue) alignée avec le plan de l'appartement

Figure 4.12 – Reconstruction d'un appartement

activités de la vie quotidienne (voir Fig.4.13a). Ainsi, si nous parvenons à localiser la caméra, alors nous aurons également la position de la personne dans l'environnement. Dans le contexte du projet européen Dem@Care par exemple, cette information est utilisée pour effectuer le suivi d'activités d'une personne atteinte de la maladie d'Alzheimer.

La difficulté de ce problème de localisation réside dans le fait que :

1. les objets avec lesquels la personne interagit sont fréquemment interposés entre la caméra et l'environnement,
2. l'image est souvent floue à cause du mouvement de la personne et de fortes différences d'illumination sont présentes,
3. l'environnement change entre le moment où l'on a pris la vidéo d'apprentissage dans le but de cartographier l'environnement et le moment où la personne effectue ses activités. De plus, les points de vue des images présentes dans la base d'images et des images de la vidéo à localiser peuvent être différents.

Ainsi, nous souhaitons ici développer une méthode de localisation qui :

- repose exclusivement sur les images provenant de la caméra portée, c'est à dire qu'aucun autre capteur (centrale inertielle, lidar, etc.) ne doit être utilisé,
- estime la position de la caméra, avec une précision inférieure à un mètre, ainsi que son orientation avec un temps de calcul "raisonnable",
- est cohérente vis à vis de la topologie de l'environnement, c'est à dire que la trajectoire de la caméra ne doit pas traverser les murs,
- est capable de détecter lorsque l'information provenant des images n'est pas suffisante pour estimer la localisation, c'est à dire quand la distribution a posteriori de la trajectoire de la caméra est soit multimodale soit trop diffuse.

Dans ce contexte, nous proposons une nouvelle méthode de VIL permettant de satisfaire les précédentes spécifications :

1. nous formulons le problème de localisation comme un problème de suivi de cible sur le groupe de Lie $SE(3)$, où la cible est la pose de la caméra. Pour chaque image de la vidéo, nous appliquons un algorithme de recherche par le contenu (CBIR en anglais pour Content Based Image Retrieval) à la base d'images, afin de retrouver les images clés de la séquence d'apprentissage les plus proches (en terme d'apparence) de l'image courante. Les coordonnées (poses) de ces images clés sont utilisées comme observations par notre algorithme de suivi de cible.

2. dans le but de résoudre ce problème de suivi de cible, tout en sélectionnant l'observation pertinente à chaque instant, nous proposons d'appliquer un lisseur particulière Rao-Blackwellisé sur groupe de Lie, qui utilise à la fois le LG-EKF et le LG-RTS présentés dans le chapitre précédent.
3. finalement, afin de prendre en compte la topologie de l'environnement, nous introduisons des "mesures virtuelles" qui permettent de guider les particules et les empêchent de traverser les murs.

4.3.2 État de l'art

Il existe plusieurs manières d'aborder le problème VIL. Nous les présentons ci-après.

Dans le domaine de la reconnaissance de lieux, des approches comme celle proposée dans [Dovgalecs 2011] appliquent des algorithmes d'apprentissage automatique pour apprendre l'apparence visuelle des lieux. Bien qu'étant robustes, ces approches, qui permettent de reconnaître la pièce dans laquelle se situe la caméra, ne sont pas capables de fournir la position et l'orientation de la caméra avec le niveau de précision désiré.

Parmi les méthodes permettant d'atteindre le niveau de précision souhaité, diverses approches ont été proposées.

Tout d'abord, ce problème peut être vu comme un problème de VSLAM monoculaire avec la connaissance supplémentaire de la reconstruction obtenue avec la vidéo d'apprentissage. Ainsi, les approches récentes de VSLAM monoculaire telles que [Engel 2014] ou [Eade 2008] pourraient être employées. L'approche de VSLAM monoculaire présentée dans le chapitre précédent pourrait bien entendu être utilisée, mais le temps de calcul serait prohibitif (5h de calculs pour une vidéo de 20 minutes).

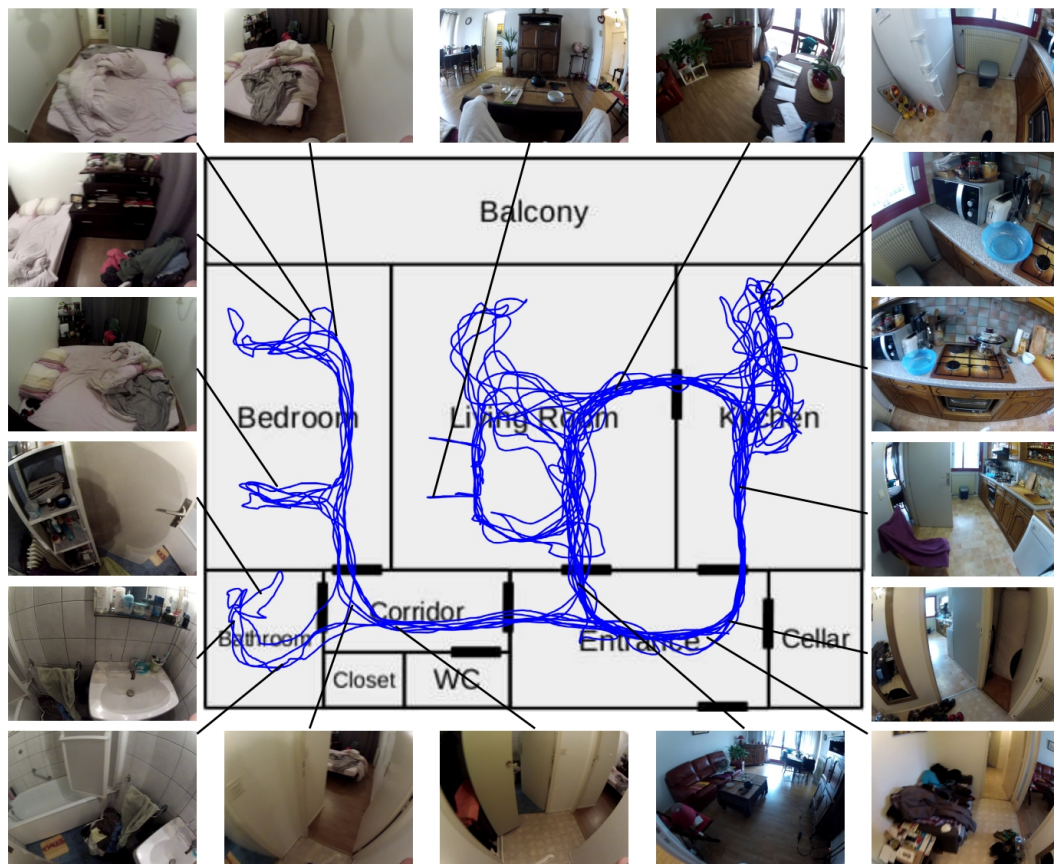
Une autre manière d'aborder le problème est de considérer le nuage de points 3D de l'environnement (par exemple issu de notre algorithme de VSLAM monoculaire). En utilisant ce nuage de points 3D, il est possible de localiser les images de la vidéo comme proposé dans [Sattler 2012], [Sattler 2011], [Liang 2013] ou [Kroeger 2014].

Finalement, les approches développées dans le domaine du CBIR [Conaire 2009 ; Lourenço 2012] et VSLAM par l'apparence [Cummins 2008] sont capables de retrouver efficacement les plus proches voisins d'une image dans la base d'images. Ainsi, les coordonnées des images retrouvées peuvent être interprétées comme la pose de l'image requête dans l'environnement.

Cependant, l'ensemble des approches précédemment citées ont été développées pour traiter des vidéos "propres" (pas d'objets devant la caméra, pas de flou de bougé, etc.). Par conséquent, ces méthodes ne sont pas capables



(a) Exemples d'images provenant de la caméra portée lorsque la personne effectue des activités de la vie quotidienne. Les objets avec lesquels la personne interagit apparaissent fréquemment dans le champs de vision de la caméra.



(b) Illustration de la base d'images d'un appartement où chaque image a été annotée avec des coordonnées (ici "coordonnées" signifie "pose de caméra", c'est à dire une position 3D et une orientation 3D). Cette base de données a été générée automatiquement à partir d'une séquence d'apprentissage comme expliqué dans la partie 4.3.3.1. La ligne bleue correspond à la trajectoire de la caméra lors de la séquence d'apprentissage projetée sur le plan de l'appartement.

Figure 4.13 – Haut : Exemples d'images à localiser. Bas : Illustration de la base d'images représentant l'environnement préalablement cartographié.

de localiser correctement la caméra dans les conditions difficiles d'une caméra portée durant des activités de la vie quotidienne.

En effet, les approches de VSLAM monoculaire sont encore fragiles et nécessitent des conditions spécifiques telles que de larges translations de la caméra, une illumination constante et un environnement statique, qui ne sont pas respectées dans notre contexte.

Les approches de localisation basées sur le nuage de points 3D supposent que l'environnement ne change pas ou peu entre le moment où l'on a effectué la vidéo d'apprentissage et celui où l'on capture nos vidéos à localiser. De plus, le nuage de points 3D est généralement supposé parfaitement estimé. Ces hypothèses sont loin d'être respectées dans un appartement habité (voir Fig.4.13a et 4.14).

Les algorithmes de recherche d'images quant à eux retrouvent souvent des images aberrantes, d'une part à cause des points de vue qui diffèrent entre la vidéo d'apprentissage et la vidéo à localiser et d'autre part à cause des objets avec les lesquels la personne interagit qui passent devant la caméra (voir Fig.4.14).

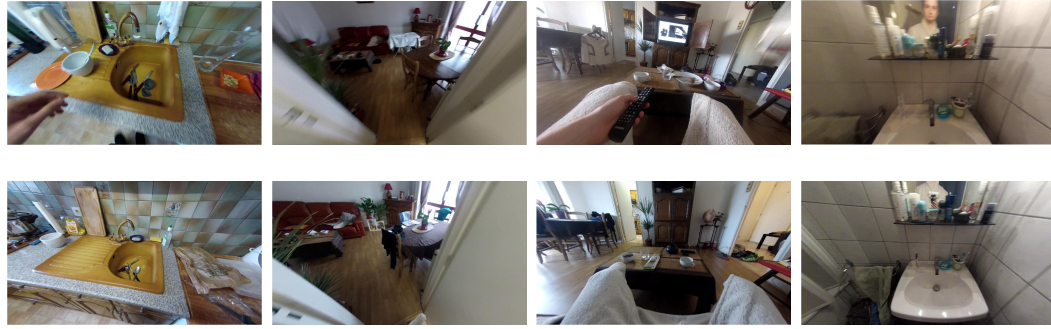
Dans ce sous-chapitre, nous proposons une méthode qui permet de contourner les limites des approches de l'état de l'art. Notre approche consiste à employer un algorithme CBIR sur les images de la vidéo à localiser, puis à utiliser les coordonnées des images retrouvées dans la base d'images comme observations dans un problème de suivi de cible. L'originalité de cette méthode repose principalement sur le fait que, dans ce problème, à la fois les observations et la trajectoire de la caméra évoluent sur le groupe de Lie $SE(3)$.

Les coordonnées aberrantes fournies par le CBIR sont rejetées en tirant profit du fait que :

- la caméra doit avoir une trajectoire continue et lisse au cours du temps,
- la caméra évolue dans un environnement intérieur contraint et par conséquent ne peut pas traverser les murs et doit emprunter les portes pour passer d'une pièce à l'autre.

Remarquons que les approches décrites dans [Ham 2005], [Quigley 2010] et [Wolf 2005] sont proches de celle que nous proposons puisqu'elles consistent également à post-traiter la sortie d'un algorithme CBIR. Par exemple, [Ham 2005] utilise un filtre Kalman alors que [Quigley 2010] et [Wolf 2005] emploient un filtre particulier pour estimer la trajectoire de la caméra à partir de la sortie d'un algorithme CBIR. Cependant, toutes ces méthodes nécessitent d'autres capteurs en plus de la caméra, estiment la trajectoire de la caméra dans un plan 2D et ne prennent pas en compte le fait que l'environnement est contraint.

Au contraire, notre méthode repose exclusivement sur l'information apportée par les images de la caméra. De plus, afin d'estimer la trajectoire de la caméra en



(a) Exemples d'images retrouvées avec succès dans la base d'images par l'algorithme CBIR. Haut : Images requêtes Bas : Images retrouvées



(b) Exemples d'images aberrantes retrouvées dans la base d'images par l'algorithme CBIR. Haut : Images requêtes Bas : Images retrouvées

Figure 4.14 – Illustration des images retrouvées par l'algorithme CBIR (plus proche voisin). (a) exemples d'images retrouvées avec succès. (b) exemples de résultats aberrants.

6 dimensions (groupe de Lie $SE(3)$) à partir d'observations évoluant elles-mêmes sur $SE(3)$, nous proposons d'employer un lisseur particulier Rao-Blackwellisé sur groupe de Lie et d'introduire des “mesures virtuelles” dans le but de guider les particules et de les empêcher de traverser les murs.

4.3.3 Architecture du système de localisation en environnement intérieur

L'architecture du système de localisation contient deux modules : un algorithme CBIR qui pour une image de la vidéo retourne les coordonnées des images visuellement proches retrouvées dans une base d'images, suivi d'un lisseur particulier Rao-Blackwellisé sur groupe de Lie (LG-RBPS en anglais pour Rao-Blackwellized Particle Smoother on Lie Groups), comme présenté en figure 4.15.

Nous présentons maintenant brièvement la manière dont nous construisons automatiquement la base d'images ainsi que l'algorithme CBIR. La partie suivante est consacrée au LG-RBPS.

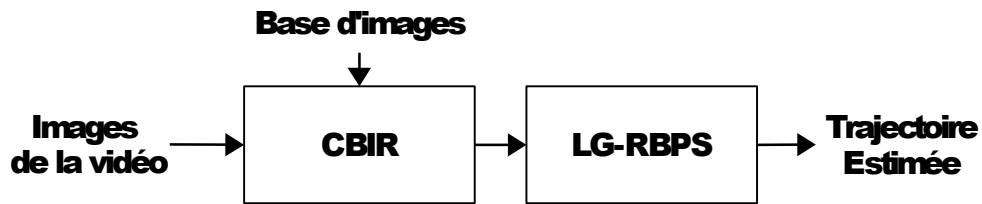


Figure 4.15 – Architecture du système de localisation en environnement intérieur

4.3.3.1 Construction automatique de la base d'images

La construction manuelle d'une base d'images annotées avec des coordonnées représentant un environnement intérieur est généralement une étape fastidieuse et peu précise. Nous proposons ici une méthode automatique de création de cette base d'images. Tout d'abord, une vidéo d'apprentissage tournée dans l'environnement et aussi exhaustive que possible est enregistrée. Ensuite, nous utilisons l'algorithme de VSLAM monoculaire présenté dans la partie 4.2. Finalement, la trajectoire estimée de la caméra est manuellement alignée avec le plan 2D du bâtiment. De cette manière, chaque image de la séquence d'apprentissage est automatiquement annotée avec des coordonnées, c'est à dire avec une position 3D et une orientation 3D. Un exemple de base d'images d'un appartement automatiquement générée à partir d'une séquence d'apprentissage de 20 minutes est présenté en figure 4.13b.

4.3.3.2 Algorithme de recherche par le contenu

L'algorithme CBIR que nous utilisons est inspiré de [Liang 2013] et [Conaire 2009] mais possède tout de même quelques différences importantes. Nous décrivons maintenant cet algorithme pour une image de la vidéo à localiser.

Tout d'abord, une image miniature de taille 32×32 est créée [Torralba 2008] et comparée aux images miniatures de la base d'images en utilisant la somme des distances absolues. A la fin de cette étape initiale, seules les 100 images retrouvées les plus proches sont conservées.

Ensuite, les points d'intérêts SURF [Bay 2006] sont détectés et appariés aux points SURF des 100 images retrouvées en utilisant un arbre-kd. A la fin de cette étape, seules les 5 images de la base d'images ayant le plus grand nombre d'appariements sont conservées.

Finalement, parmi ces 5 images, seules celles ayant au moins 15 appariements sont conservées. Ainsi, pour une image requête, l'algorithme CBIR que nous utilisons fournit au maximum 5 images de la base d'images, générant ainsi au maximum 5 mesures appartenant au groupe de Lie $SE(3)$.

Afin de réduire le coût calculatoire de l'algorithme CBIR, nous l'utilisons uniquement sur une image par seconde de la vidéo à localiser.

Des exemples de résultats produits par cet algorithme sont présentés en figure 4.14 (seul le plus proche voisin au sens du nombre d'appariements est présenté).

Cet algorithme CBIR permet de générer efficacement des hypothèses de poses pour une image de la vidéo.

4.3.4 Un problème de suivi de cible sur $SE(3)$

Après avoir obtenu des coordonnées possibles des images de la vidéo, en utilisant un algorithme CBIR, nous souhaitons maintenant estimer la trajectoire de la caméra.

Pour cela, nous formulons ce problème d'estimation de trajectoire comme un problème de suivi de cible sur le groupe de Lie $SE(3)$ où les observations que nous considérons sont les coordonnées (vivant sur le groupe de Lie $SE(3)$) des images retrouvées par l'algorithme CBIR. De plus, afin de prendre en compte le fait que la trajectoire de la caméra ne doit pas traverser et emprunter les portes, nous proposons d'introduire des "observations virtuelles". Pour résoudre ce problème de suivi de cible, nous déployons un nouvel algorithme de lissage particulier Rao-Blackwellisé sur groupe de Lie.

4.3.4.1 Formulation du problème

Le modèle graphique associé au problème de suivi de cible que nous considérons est présenté en figure 4.16. Comme nous allons le voir plus en détails dans les paragraphes qui suivent, les variables X_k représentent la trajectoire de la caméra. L'observation Z_k contient plusieurs composantes et la variable s_k permet d'en sélectionner une. Ainsi, l'objectif de notre approche est d'estimer conjointement la trajectoire de la caméra tout en sélectionnant l'observation pertinente à chaque instant.

Mesures virtuelles L'algorithme CBIR retrouve fréquemment des images aberrantes dans la base d'images, ce qui donne lieu à des coordonnées aberrantes (voir Fig.4.14). Par conséquent, nous devons trouver une manière de rejeter les coordonnées fournies par l'algorithme CBIR quand cela arrive.

De plus, pour obtenir une trajectoire de caméra cohérente vis à vis de l'environnement dans lequel elle évolue, il est nécessaire de "guider" la pose de la caméra pour l'empêcher de traverser les murs et la forcer à passer par les portes pour passer d'une pièce à l'autre.

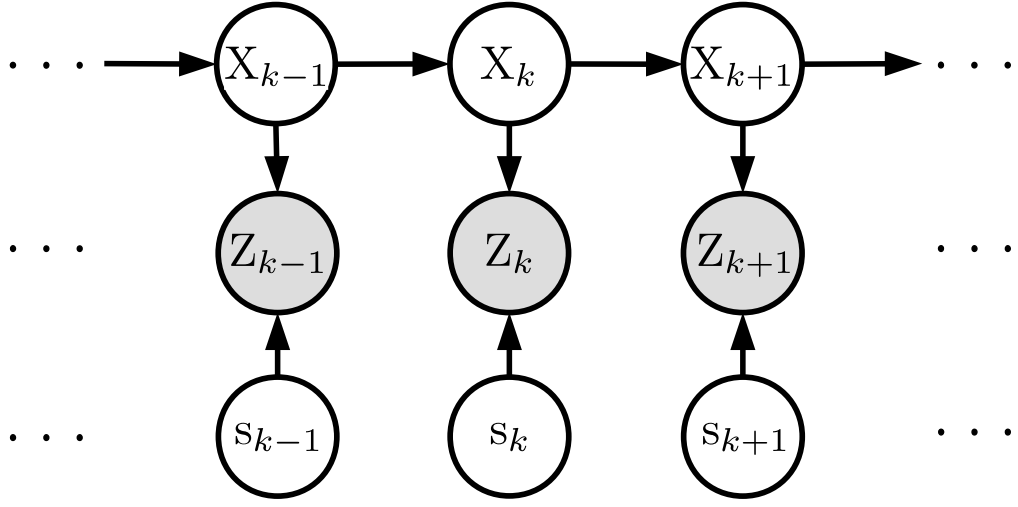


Figure 4.16 – Modèle graphique du problème de localisation

Dans le but de répondre à ces deux problèmes, nous définissons les observations Z_k disponibles à l'instant k de la manière suivante :

$$Z_k = \left[\underbrace{Z_k(1), Z_k(2), \dots, Z_k(N)}_{\text{Mesures Virtuelles}}, \underbrace{Z_k(N+1), \dots, Z_k(N+N_{CBIR})}_{\text{Sortie CBIR}} \right] \quad (4.13)$$

Les N premières composantes de Z_k sont les coordonnées de toutes les images présentes dans la base d'images. Nous appelons ces composantes des “mesures virtuelles”.

Les N_{CBIR} composantes suivantes sont les coordonnées des images de la base d'images retrouvées par l'algorithme CBIR. Dans notre cas N_{CBIR} vaut au maximum 5 car l'algorithme CBIR retourne au maximum 5 images (voir partie 4.3.3.2).

Ainsi, à chaque instant les observations Z_k appartiennent au groupe de Lie $(SE(3))^{N+N_{CBIR}}$ 6.

Remarquons que les composantes correspondant à la sortie du CBIR sont dupliquées puisqu'elles sont déjà présentes dans les mesures virtuelles. Une explication de cette duplication est donnée à la fin de cette sous-partie.

Un des objectifs de notre algorithme est de sélectionner, à chaque instant, la composante qui semble la plus vraisemblable parmi les $N + N_{CBIR}$ composantes disponibles.

6. La représentation utilisée dans (4.13) est un peu abusive et doit être interprétée symboliquement. En effet, Z_k correspond à $N + N_{CBIR}$ produits directs du groupe de Lie $SE(3)$ et devrait être représenté par une matrice de taille $6(N + N_{CBIR}) \times 6(N + N_{CBIR})$.

Variables de sélection Dans le but de sélectionner l'observation pertinente à chaque instant, nous introduisons une variable de sélection $s_k \in \{1, \dots, N + N_{CBIR}\}$ qui agit comme un sélecteur parmi les composantes de Z_k , c'est à dire que $Z_k(s_k = i)$ sélectionne la i ème composante de Z_k .

La probabilité de transition des variables de sélection s_k est définie comme :

$$p(s_k | s_{k-1}) = p(s_k) \quad (4.14)$$

Ainsi, ces probabilités de transition peuvent être vues comme les poids des composantes de Z_k . En pratique, $p(s_k = i) \ll p(s_k = j)$ pour $i \leq N$ et $j > N$ de manière à encourager le système à “utiliser” le plus possible les coordonnées fournies par l'algorithme CBIR.

Modèle de propagation Afin d'estimer conjointement la trajectoire de la caméra tout en sélectionnant l'observation pertinente à chaque instant, nous souhaitons ajouter l'information traduisant le fait que la trajectoire de la caméra doit être continue et lisse. Nous proposons donc d'utiliser le modèle à accélération blanche déjà employé dans la partie 3.6.6 :

$$\begin{cases} C_{k+1} &= \exp_{SE(3)}^\wedge(v_k \Delta t) C_k \\ v_{k+1} &= v_k + n_k \end{cases} \quad (4.15)$$

où $C_k \in SE(3)$ représente la pose de la caméra à l'instant k . Plus précisément, C_k est défini comme la transformation $C_{i_k S}$ où S est le référentiel absolu de la carte et i_k est le référentiel de la caméra à l'instant k . $v_k \in \mathbb{R}^6$ correspond à la vitesse de la caméra, $n_k \sim \mathcal{N}_{\mathbb{R}^6}(n_k; \mathbf{0}_{6 \times 1}, R_k)$ est un bruit blanc Gaussien et Δt correspond à l'intervalle de temps entre deux images de la vidéo. Ainsi, l'état $X_k \in SE(3) \times \mathbb{R}^6$ que nous souhaitons estimer à l'instant k est le produit direct de la pose de la caméra C_k et de sa vitesse v_k .

Vraisemblance Nous considérons la vraisemblance suivante :

$$p(Z_k | C_k, s_k = i) = \prod_{j=1}^{N+N_{CBIR}} p(Z_k(j) | C_k, s_k = i) \propto \mathcal{N}_{SE(3)}^R(Z_k(i); C_k, Q_{k,i}) \quad (4.16)$$

où $Q_{t,i} = Q_{VM}$ si $i \leq N$ et $Q_{t,i} = Q_{CBIR}$ sinon. Les matrices de covariance Q_{VM} et Q_{CBIR} sont supposées connues. Dans la définition de la vraisemblance (4.16), nous avons supposé que les composantes de Z_k sont conditionnellement indépendantes les unes des autres étant donné la pose de la caméra et la variable de sélection. De plus, seule la composante sélectionnée apporte une information,

les autres composantes sont supposées avoir une distribution non informative d'où la relation de proportionnalité dans (4.16).

Remarquons que le fait que les composantes de Z_k issues de l'algorithme CBIR soient également présentes parmi les composantes des mesures virtuelles ne pose aucun problème. Une manière (quasi) équivalente de présenter les choses serait de supprimer les composantes issues du CBIR et de reporter leurs poids sur les composantes des mesures virtuelles correspondantes. Ainsi, on pourrait interpréter l'approche proposée sous un nouvel angle où, à chaque instant, l'algorithme doit choisir la pose d'une des images de la base d'images alors que l'information issue du CBIR est prise en compte au niveau des poids (valeurs des $p(s_k)$).

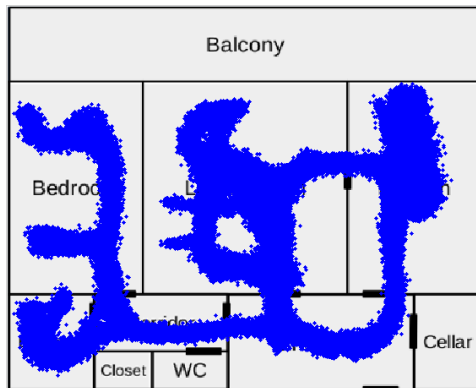
Cependant, cette manière de voir les choses ne permet pas de spécifier des covariances différentes pour les mesures virtuelles Q_{VM} et les mesures issues de l'algorithme CBIR Q_{CBIR} comme nous l'avons fait.

Explications Tout d'abord, en introduisant les mesures virtuelles et les variables de sélection, nous avons rendu le système capable de sélectionner une mesure virtuelle au lieu d'une observation issue du CBIR lorsque ce dernier ne retrouve que des données aberrantes. Donc, de ce point de vue, le système est désormais "robuste" aux données aberrantes.

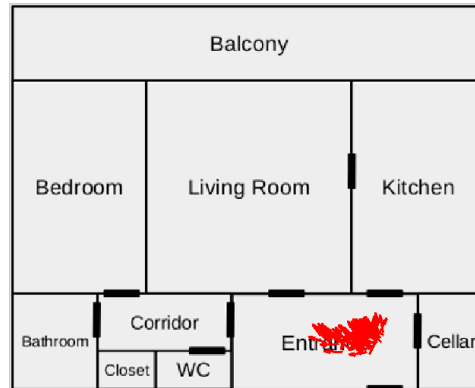
Ensuite, le fait d'avoir introduit des mesures virtuelles avec un modèle de mouvement à accélération blanche permet de guider les particules lorsque l'algorithme CBIR ne fonctionne pas. En effet, à chaque pas de temps, le système doit alors sélectionner une mesure virtuelle, ce qui le force à ne pas traverser les murs, à suivre les couloirs et à emprunter les portes (voir Fig.4.17). Bien entendu, cette astuce empêche les particules de "traverser" les murs uniquement si la vitesse de la caméra reste faible et que le pas de temps Δt est petit. Si ces deux conditions ne sont pas respectées, une particule peut faire de grands sauts à chaque pas de temps et ainsi passer d'une pièce à l'autre en traversant un mur par exemple. En pratique, nous forçons simplement numériquement la vitesse de la caméra à ne pas dépasser un seuil physiquement réaliste et Δt vaut la durée entre deux images de la vidéo, c'est à dire $\frac{1}{25}$ sec, ce qui ne permet pas à la caméra de parcourir une grande distance.

4.3.4.2 Lisseur particulaire Rao-Blackwellisé sur groupe de Lie

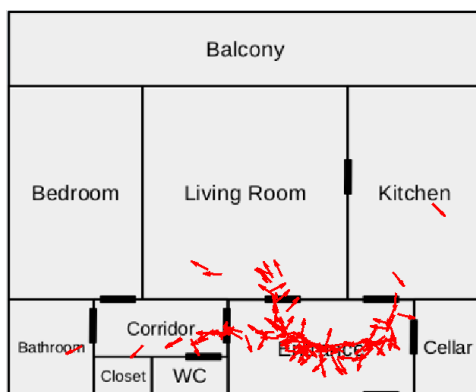
Afin de satisfaire les spécifications techniques décrites précédemment (voir partie 4.3.1), c'est à dire estimer non seulement la trajectoire de la caméra mais également avoir une notion de l'incertitude de cette trajectoire estimée, nous avons choisi d'approcher la distribution a posteriori complète



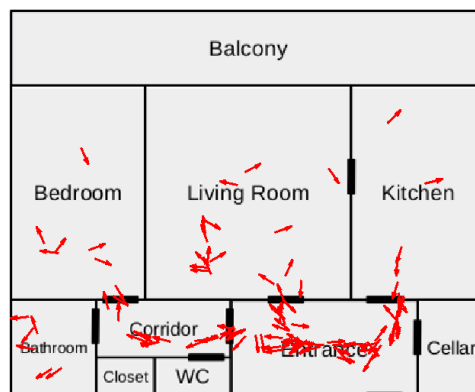
(a) 20 échantillons de chaque mesure virtuelle sont ici représentés. La zone formée par tous ces échantillons représente donc l'ensemble des endroits où la caméra est "autorisée" à aller.



(b) $t=0.3$: Dans cet exemple, les particules sont toutes initialisées avec la même pose de caméra et une vitesse nulle.



(c) $t=3$ sec : Les particules commencent à se propager dans les pièces par les portes ainsi que le long du couloir.



(d) $t=6$ sec : Les particules ont maintenant atteint toutes les pièces.

Figure 4.17 – Illustration de la diffusion des particules en utilisant uniquement les mesures virtuelles (l'algorithme CBIR est ici désactivé). Les positions 3D des caméras sont ici projetées (bases des flèches) sur le plan de l'appartement ainsi que leurs orientations (directions des flèches).

$p(X_0, X_1, \dots, X_T, s_1, s_2, \dots, s_T | Z_1, Z_2, \dots, Z_T) = p(X_{0:T}, s_{1:T} | Z_{1:T})$ où T est la durée de la vidéo à localiser.

Le problème que nous avons formulé (voir paragraphe 4.3.4.1) présentant des variables continues ainsi que des variables discrètes, nous proposons de le résoudre en utilisant une méthode de Monte-Carlo.

Cependant, si la séquence $s_{1:T}$ est supposée connue, l'estimation de $X_{0:T}$ peut être réalisée par les algorithmes LG-EKF et LG-RTS introduits au chapitre 3. C'est le principe de la Rao-Blackwellisation.

Ainsi, dans cette sous-partie, nous proposons d'employer un lisseur particulaire Rao-Blackwellisé sur groupe de Lie qui permet de tirer parti de la structure du problème ((4.15), (4.16) et 4.14) pour échantillonner efficacement l'espace d'état et ainsi utiliser "peu" de particules⁷.

État de l'art Dans le domaine de la localisation et cartographie simultanée, plusieurs travaux ont proposé des lisseurs particuliers Rao-Blackwellisé (RBPS en anglais pour Rao-Blackwellized Particle Smoother) [Beever 2007]. Néanmoins, ces approches sont des RBPS "classiques", c'est à dire que le problème considéré conduit simplement à l'utilisation des algorithmes IEKF et RTS euclidiens. De plus, les observations considérées n'évoluent pas sur un groupe de Lie.

Dans [Kwon 2010], un filtre particulaire Rao-Blackwellisé (RBPF en anglais pour Rao-Blackwellized Particle Filter) capable de traiter des observations évoluant sur un groupe de Lie est proposé. Bien qu'une transformée sans parfum sur groupe de Lie soit proposée, le problème considéré ne conduit pas les auteurs à proposer un RBPF sur groupe de Lie impliquant un filtre de Kalman sur groupe de Lie.

Dans [Barrau 2014], un RBPF invariant est proposé. Cet algorithme est dédié à des systèmes possédant des symétries conditionnellement à certaines variables de l'état. Ainsi, les auteurs montrent qu'il est possible de n'échantillonner qu'une partie de l'état et d'employer le filtre de Kalman étendu invariant [Bonnabel 2009] pour estimer l'autre partie de l'état.

Contrairement à ces approches, nous proposons un algorithme LG-RBPS capable à la fois :

- de prendre en compte le fait que les observations évoluent sur un groupe de Lie

7. Réduire le nombre de particules est essentiel lorsque l'état évolue sur un groupe de Lie. En effet, générer des particules sur groupe de Lie nécessite l'utilisation de l'exponentielle de matrice, ce qui implique un coût calculatoire important par rapport à des particules "euclidiennes" [Berger 2015].

- de tirer parti de la structure du problème considéré pour proposer une étape de filtrage n'échantillonnant qu'une partie de l'état et employant l'algorithme LG-IEKF pour gérer l'autre partie
- de tirer parti de la structure du problème considéré pour proposer une étape de lissage utilisant le résultat de l'étape de filtrage et impliquant l'algorithme LG-RTS pour échantillonner efficacement l'état complet

Ainsi, l'approche que nous proposons peut être vue comme une généralisation du RBPS de [Fong 2002] aux groupes de Lie.

Algorithme LG-RBPS

Obtention des équations du LG-RBPS La démonstration conduisant à l'algorithme LG-RBPS est présentée en annexe H. Le pseudo-code du LG-RBPS est présenté Alg.4.3 et une description des différentes étapes est fournie ci-après.

Explications L'algorithme LG-RBPS présenté dans l'Alg.4.3, qui permet d'estimer la trajectoire de la caméra, consiste en deux étapes : une étape de filtrage suivi d'une étape de lissage (qui repose sur le résultat de l'étape de filtrage).

Dans l'étape de filtrage, l'objectif est d'approcher la distribution de probabilité suivante :

$$p(X_{0:k}, s_{0:k} | Z_{1:k}) = p(X_{0:k} | Z_{1:k}, s_{0:k}) p(s_{0:k} | Z_{1:k}) \quad (4.17)$$

Le terme $p(s_{0:k} | Z_{1:k})$ est échantillonné par N_p particules de la manière suivante (tirées selon une distribution de proposition $q(s_{0:k} | Z_{1:k})$) :

$$p(s_{0:k} | Z_{1:k}) \approx \sum_{l=1}^{N_p} w_k^{(l)} \delta(s_{0:k} - s_{0:k}^{(l)}) \quad (4.18)$$

Le terme $p(X_{0:k} | Z_{1:k}, s_{0:k})$ est approché par une distribution normale concentrée sur groupe de Lie en utilisant le filtre de Kalman étendu sur groupe de Lie (LG-EKF) présenté dans le chapitre 3. L'efficacité de cet algorithme réside dans le fait que seul $s_{0:k}$ est échantillonné, c'est à dire que $X_{0:k}$ a été marginalisé dans (4.18). Sachant $s_{0:k}$, la partie $X_{0:k}$ de l'état marginalisé peut alors être estimée analytiquement (LG-EKF). Ainsi, N_p trajectoires pondérées sont générées récursivement.

Une trajectoire est paramétrée par sa moyenne $\mu \in SE(3) \times \mathbb{R}^6$, sa covariance $P \in Sym^+(12)$ ainsi qu'un poids w à chaque instant. Nous expliquons maintenant comment une de ces trajectoires est obtenue.

La moyenne de la trajectoire est initialisée en tirant aléatoirement une pose de caméra parmi les poses des images présentes dans la base d'images avec une vitesse nulle. La covariance de la trajectoire est initialisée avec une matrice de covariance prédéfinie P_0 et la valeur initiale de w est $\frac{1}{N_p}$.

A chaque instant k , allant de $k = 1$ (première image de la vidéo) à $k = T$ (dernière image de la vidéo), l'algorithme fonctionne de la manière suivante :

1. Étape de propagation du LG-EKF : la moyenne et la covariance de la trajectoire sont propagées en utilisant le modèle de propagation (4.15)
2. Échantillonnage optimal : la vraisemblance de chaque composante de Z_k , étant donné la trajectoire propagée, est calculée et utilisée pour tirer une des composantes
3. Étape de mise à jour du LG-EKF : la moyenne et la covariance de la trajectoire propagée sont corrigées (en utilisant le modèle d'observation (4.16)) en incorporant l'information contenue dans la composante de Z_k sélectionnée à l'étape précédente
4. Mise à jour du poids de la trajectoire en sommant les vraisemblances de toutes les composantes de Z_k calculées à l'étape 2

Après avoir appliqué ces quatre étapes à chacune des N_p trajectoires, les poids sont normalisés et les trajectoires peu vraisemblables (faible poids) sont remplacées⁸ par des trajectoires plus vraisemblables (fort poids) lors de l'étape de ré-échantillonnage.

Les calculs en lien avec cette étape de filtrage sont présentés en annexe H.1.

Les trajectoires produites par l'étape de filtrage peuvent contenir des "sauts" en raison de l'étape de ré-échantillonnage. De plus, à chaque instant, seule l'information venant du passé est utilisée pour tirer une composante parmi les composantes de l'observation courante. Or, l'information future contient une quantité d'information qui peut s'avérer très importante, surtout dans des situations où les images récemment passées sont peu informatives (mains devant la caméra, changement d'illumination, etc.). L'étape de lissage du LG-RBPS permet de prendre en compte cette information.

Dans l'étape de lissage, l'algorithme marche à rebours, c'est à dire de $k = T$ jusqu'à $k = 1$, et génère N_p échantillons de la distribution a posteriori $p(X_{0:T}, s_{1:T} | y_{1:T})$. Par souci de simplicité, nous supposons ici que l'algorithme génère autant de trajectoires à l'étape de filtrage qu'à l'étape de lissage, mais rien n'empêche de réduire ce nombre pour la seconde étape, surtout que le lissage d'une particule est coûteux calculatoirement. De plus, l'algorithme peut être exécuté par bloc, ce qui permet de réduire l'empreinte mémoire de

8. Ici "remplacer" signifie que l'on conserve l'historique de la trajectoire et on remplace simplement la moyenne et la covariance courante de la trajectoire par celles d'une autre trajectoire.

l'algorithme [Fong 2002]. Remarquons que, contrairement à l'étape de filtrage où seul s_k est échantillonné, ici toutes les variables (X_k et s_k) le sont.

Pour obtenir chaque échantillon, l'algorithme construit (à rebours) une trajectoire. L'algorithme débute en tirant une des trajectoires pondérées produites par l'étape de filtrage et essaye récursivement de construire une trajectoire probable parmi les trajectoires possibles produites par l'étape de filtrage. A chaque pas de temps, l'algorithme LG-RTS est utilisé pour fusionner l'information venant du passé (contenue dans la trajectoire sélectionnée issue de l'étape de filtrage) avec l'information venant du futur (contenue dans la trajectoire que l'on crée à rebours).

Les calculs en lien avec cette étape de lissage sont présentés en annexe H.2.

Finalement, pour chaque instant k , le barycentre et la covariance de l'ensemble des échantillons générés sont calculés.

Les barycentres sont utilisés comme estimée de la trajectoire de la caméra, alors que les covariances peuvent être employées pour détecter les moments où la distribution a posteriori de la trajectoire de la caméra à un instant donné est multi-modale ou trop dispersée.

4.3.5 Limites de la méthode proposée

La méthode de localisation proposée dans ce chapitre présente plusieurs limites.

Tout d'abord, la base d'images doit être aussi complète que possible. En effet, l'algorithme CBIR fonctionne uniquement si la caméra se situe dans un endroit préalablement cartographié. En pratique, la méthode de création automatique de la base d'images (voir partie 4.3.3.1) permet de créer très simplement des bases d'images grandes et complètes.

Ensuite, l'introduction de mesures virtuelles est censée empêcher les particules de traverser les murs. En théorie, une particule ayant une grande vitesse pourrait traverser un mur et atteindre une mesure virtuelle de l'autre côté. En pratique, nous utilisons un seuil afin de contraindre la vitesse de caméra à avoir une vitesse physiquement réaliste ainsi qu'un petit pas temps $\Delta t = \frac{1}{25}$ sec pour que cela n'arrive pas.

Finalement, la méthode repose essentiellement sur le fait que l'algorithme CBIR est capable de retrouver "régulièrement" des images non aberrantes dans la base d'images. En effet, le LG-RBPS effectue un post-traitement du résultat du CBIR. Si ce dernier ressort une quantité trop importante de résultats aberrants, le post-traitement ne permettra pas d'obtenir une trajectoire cohérente de la caméra.

Algorithme 4.3 LG-RBPS

Entrée : N_p (nombre de particules), P_0 (covariance initiale), T (nombre de pas de temps), $\{Z_k\}_{k=1,\dots,T}$ (observations), Q_{VM} (covariance des mesures virtuelles), Q_{CBIR} (covariance des mesures CBIR), Δt (intervalle de temps entre deux images de la vidéo), R_k (covariance du bruit de modèle), $\{p(s_k)\}_{k=1,\dots,T}$ (distribution de probabilité discrète)

Sorties : $\{\mu_k\}_{k=1,\dots,T}$ (trajectoire estimée), $\{P_k\}_{k=1,\dots,T}$ (covariance de la trajectoire estimée)

Notations : $\mu_{t_a|t_b}^{(i)}$ et $P_{t_a|t_b}^{(i)}$ correspondent à la moyenne et la covariance de la trajectoire associée à la particule i , au temps $k = t_a$, ayant observé $\{Z_k\}_{k=1,\dots,t_b}$

Étape de filtrage

Pour $i = 1, 2, \dots, N_p$

- Initialiser $\mu_{0|0}^{(i)}$ en choisissant aléatoirement la pose d'une des images de la base d'images avec une vitesse nulle
- $P_{0|0}^{(i)} = P_0$ et $w_0^{(i)} = \frac{1}{N_p}$

FinPour

Pour $k = 1, 2, \dots, T$

- Pour $i = 1, 2, \dots, N_p$
 - **Propagation du LG-EKF** : Propager $\mu_{k-1|k-1}^{(i)}$ et $P_{k-1|k-1}^{(i)}$ pour obtenir $\mu_{k|k-1}^{(i)}$ et $P_{k|k-1}^{(i)}$ en utilisant R_k et Δt
 - **Échantillonnage optimal** : Tirer $s_k^{(i)}$ en utilisant $\mu_{k|k-1}^{(i)}$, $P_{k|k-1}^{(i)}$, Z_k et $p(s_k)$, et évaluer $p(Z_k | s_{0:k-1}^{(i)}, Z_{1:k-1})$
 - **Mise à jour du LG-EKF** : Mettre à jour $\mu_{k|k-1}^{(i)}$ et $P_{k|k-1}^{(i)}$ pour obtenir $\mu_{k|k}^{(i)}$ et $P_{k|k}^{(i)}$ en utilisant $y_k(s_k^{(i)})$ et Q_{CBIR} ou Q_{VM}
 - Mise à jour du poids : $w_k^{(i)} = w_{k-1}^{(i)} p(Z_k | s_{0:k-1}^{(i)}, Z_{1:k-1})$
- FinPour
- Normaliser les poids et ré-échantillonner les particules

FinPour

Étape de lissage

Pour $i = 1, 2, \dots, N_p$

- Faire $j = l$ avec probabilité $w_T^{(l)}$
- $\tilde{s}_T = s_T^{(j)}$, $\tilde{\mu}_{T|T} = \mu_{T|T}^{(j)}$, $\tilde{P}_{T|T} = P_{T|T}^{(j)}$
- Tirer $X_T^{(i)} \sim \mathcal{N}_{SE(3) \times \mathbb{R}^6}^R(X_T; \tilde{\mu}_{T|T}, \tilde{P}_{T|T})$
- Pour $k = T-1, T-2, \dots, 1$
 - Pour $l = 1, 2, \dots, N_p$
 - Donner à r la valeur de $\mathcal{N}_{SE(3) \times \mathbb{R}^6}^R(X_{k+1}; \mu_{k+1|k}^{(l)}, P_{k+1|k}^{(l)})$ évaluée en $X_{k+1} = X_{k+1}^{(i)}$
 - $w_{k|k+1}^{(l)} \propto w_k^{(l)} p(\tilde{s}_{k+1}) r$
 - FinPour
 - Faire $j = l$ avec probabilité $w_{k|k+1}^{(l)}$
 - $\tilde{s}_k = s_k^{(j)}$, $\tilde{\mu}_{k|k} = \mu_{k|k}^{(j)}$ et $\tilde{P}_{k|k} = P_{k|k}^{(j)}$
 - **Algorithme LG-RTS** : Lisser $\tilde{\mu}_{k|k}$ et $\tilde{P}_{k|k}$ pour obtenir $\tilde{\mu}_{k|T}$ et $\tilde{P}_{k|T}$ en utilisant $\tilde{\mu}_{k+1|T}$ et $\tilde{P}_{k+1|T}$
 - Tirer $X_k^{(i)} \sim \mathcal{N}_{SE(3) \times \mathbb{R}^6}^R(X_k; \tilde{\mu}_{k|T}, \tilde{P}_{k|T})$
- FinPour

FinPour

Finalement, pour $k = 1, 2, \dots, T$ calculer le barycentre μ_k et la covariance P_k des échantillons $\{X_k^{(i)}\}_{i=1,\dots,N_p}$

Les détails concernant la propagation du LG-EKF, l'échantillonnage optimal, la mise à jour du LG-EKF et le lisseur LG-RTS sont présentés en annexe [H](#).

4.3.6 Résultats

La méthode proposée de localisation en environnement intérieur est ici évaluée sur plusieurs vidéos où la personne portant la caméra effectue des activités de la vie quotidienne.

Pour toutes ces expériences, les paramètres de notre méthode ont été optimisés manuellement sur une vidéo, une fois pour toutes :

- $\Delta t = \frac{1}{25} \text{sec}$
- $Q_{VM} = \text{diag}(\pi_{1 \times 3}, (5e - 3)_{1 \times 3})$
- $Q_{CBIR} = \text{diag}(0.1_{1 \times 3}, (5e - 2)_{1 \times 3})$
- $P_0 = \text{diag}(0.5_{1 \times 3}, (5e - 2)_{1 \times 3}, \mathbf{1}_{1 \times 6})$
- $N_p = 100$
- $p(s_k = i) = 0.1$ pour $i > N$ et $p(s_k = i) = \frac{1 - 0.1 * N_{CBIR}}{N}$ pour $i \leq N$

En pratique, lors de l'étape de mise à jour des poids, le terme

$$p(Z_k | s_{0:k-1}^{(i)}, Z_{1:k-1}) = \sum_{l=1}^{N+N_{CBIR}} w(l) \quad (4.19)$$

qui représente la somme des vraisemblances de chaque composante de Z_k est modifié. En effet, la vraisemblance $w(l)$ d'une mesure virtuelle est remplacée par la constante $\left(\sqrt{(2\pi)^6 |Q_{VM}|}\right)^{-1} p(s_k = l)$. Cela permet notamment de ne pas modifier le poids des particules lorsque le CBIR ne retrouve aucune image et que les particules “se diffusent”. Sinon, lors de la “diffusion”, les poids des particules varieraient, entraînant un ré-échantillonnage que l'on ne souhaite pas à ce stade.

De plus, tous les 15 pas de temps, nous utilisons comme valeur pour la covariance du bruit de modèle $R_t = P_0$ afin de redonner de la flexibilité aux particules.

Nous avons manuellement annoté 6 vidéos différentes. Comme nous l'avons expliqué dans la partie 4.3.2, à notre connaissance, il n'existe pas d'approche de l'état de l'art capable de fonctionner correctement dans des conditions aussi difficiles. Par conséquent, pour évaluer les performances de notre approche, nous fournissons les résultats obtenus par l'algorithme CBIR seul, par le LG-RBPS sans mesure virtuelle⁹ et par le LG-RBPS avec mesures virtuelles.

Le RMSE (m) de chaque approche est présenté dans le tableau 4.1. Pour chaque vidéo, comme nous pouvions nous y attendre, le LG-RBPS (avec et

9. Dans ce cas, l'algorithme “choisit”, à chaque pas de temps, entre faire l'étape de mise à jour avec une observation du CBIR ou bien simplement de pas faire l'étape de mise à jour (voir [Särkkä 2007]).

4.3. Localisation à partir d'une caméra monoculaire dans un environnement cartographié

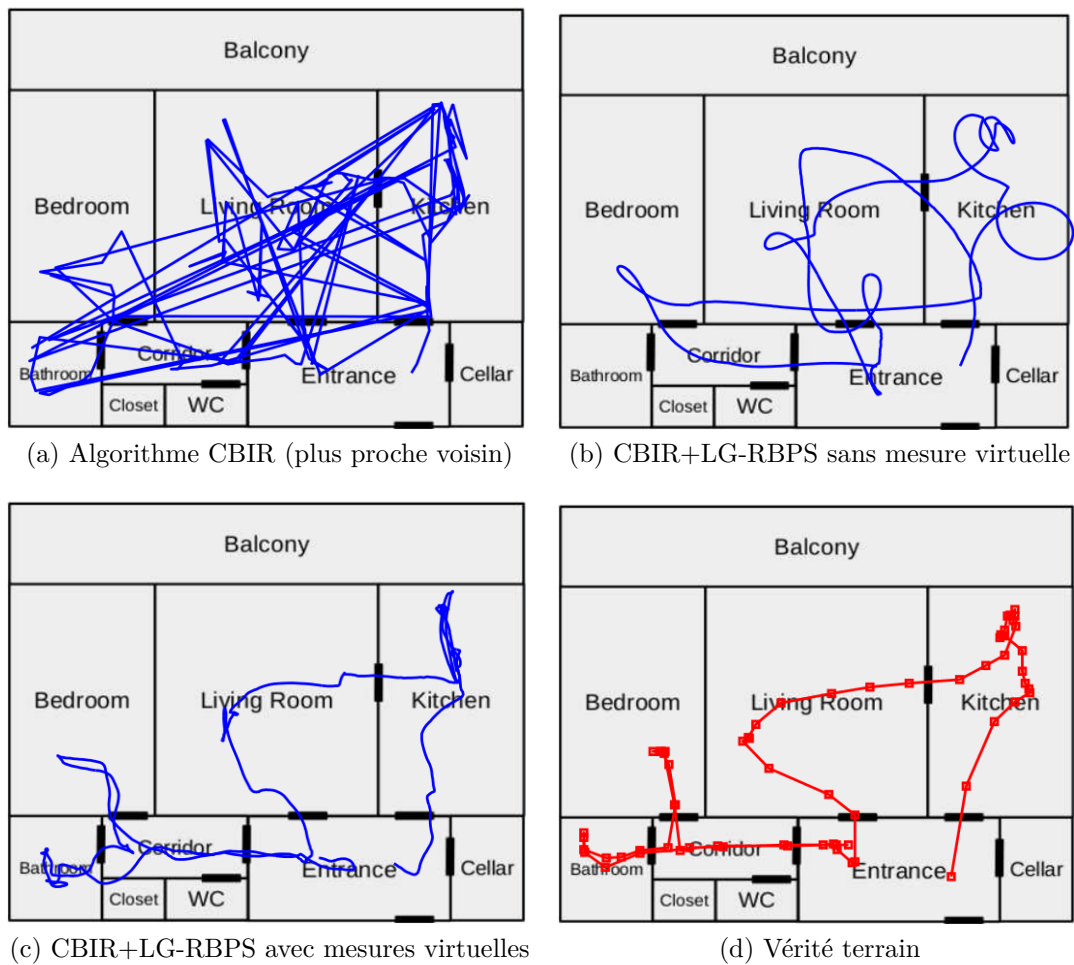


Figure 4.18 – Illustration de trajectoires estimées sur le vidéo GO82. Seule la projection de la position 3D sur le plan de l'appartement est représentée.

sans mesures virtuelles), qui vise à post-traiter la sortie du CBIR, obtient un RMSE sensiblement plus faible que celui du CBIR.

De plus, le fait d'ajouter les mesures virtuelles permet également d'améliorer les résultats, surtout lorsque le CBIR présente de "mauvais" résultats.

Dans la figure 4.18, la trajectoire de la caméra estimée avec les trois différentes approches sur la vidéo GO82 est présentée. Sur cette séquence, le LG-RBPS avec mesures virtuelles est la seule méthode pour laquelle la trajectoire estimée ne traverse pas les murs.

Un vidéo illustrant le système fonctionnant sur diverses vidéos est disponible à l'adresse suivante :

<https://sites.google.com/site/guillaumebourmaud/home/videos>

	GO80	GO81	GO82	GO83	GO84	GO85
CBIR seul	1.7	1.3	2.4	2.2	1.3	2.2
CBIR + LG-RBPS Sans Mes. Virt.	0.5	0.7	1.7	0.9	<0.5	1.3
CBIR + LG-RBPS Avec Mes. Virt.	0.5	<0.5	0.8	0.7	<0.5	0.9

Tableau 4.1 – Résultats sur 6 différentes vidéos (GO80,..., GO85). Des exemples d'images provenant de ces vidéos sont présentés en figure 4.13a et figure 4.14. Les chiffres représentent le RMSE en mètres de la trajectoire estimée par rapport à la vérité terrain (qui a une précision d'environ 0.5m).

4.4 Résumé des contributions et conclusion

Dans ce chapitre, nous avons proposé un système complet de localisation d'une caméra évoluant dans un environnement intérieur préalablement cartographié. L'approche proposée repose sur deux étapes. La première étape vise à construire une base d'images représentant l'environnement intérieur du mieux possible. Dans cette base d'images, chaque image est annotée avec une orientation 3D et une position 3D. La seconde étape utilise cette base d'images annotées pour tenter de localiser une nouvelle vidéo dans l'environnement. Dans chacune de ces deux étapes, les concepts et algorithmes d'estimation sur groupe de Lie présentés dans les chapitres précédents sont utilisés, à la fois pour obtenir une méthode élégante et numériquement stable, mais également pour caractériser l'incertitude des paramètres estimés.

Tout d'abord, afin de rendre l'étape de construction de la base d'images la plus automatique possible, nous avons introduit une méthode de localisation et de cartographie simultanée capable de cartographier un environnement intérieur à partir d'une vidéo d'apprentissage exhaustive. Afin d'obtenir une méthode robuste, nous avons notamment proposé une technique d'odométrie visuelle basée sur le Problème des Orientations Connues permettant d'estimer des sous-cartes dans des conditions difficiles. Ensuite, dans le but d'aligner les sous-cartes et d'obtenir une carte complète de l'environnement, nous avons montré une manière de détecter les fausses fermetures de boucles en utilisant un test statistique sur le groupe de Lie des similitudes 3D. Finalement, une fois les fermetures de boucles aberrantes écartées, nous avons présenté une approche de propagation de croyance permettant d'aligner efficacement un grand nombre de sous-cartes. Nous avons montré sur de nombreux jeux de données que l'approche proposée est à la fois plus robuste et précise que les algorithmes de l'état de l'art. De plus, pour toutes ces expériences, les paramètres de l'approche ont été fixés une fois pour toutes montrant la flexibilité de l'approche proposée.

Dans une seconde partie, nous avons présenté une méthode de localisation en environnement intérieur, dédiée à des vidéos provenant d'une caméra monoculaire portée où la personne portant la caméra effectue des activités de la vie quotidienne. Nous avons ainsi introduit un lisseur Rao-Blackwellisé sur groupe de Lie reposant sur les algorithmes LG-IEKF et le LG-RTS présentés dans le chapitre 3. Ce lisseur permet de post-traiter la sortie d'un algorithme de recherche par le contenu qui, pour une image de la vidéo, retrouve des images visuellement proches dans la base d'images. Nous avons montré sur plusieurs vidéos que l'approche proposée permet d'estimer la position de la caméra avec une précision de moins d'un mètre, ainsi que son orientation. A notre connaissance, il s'agit du premier système de localisation visuelle utilisant uniquement

l'information provenant d'une caméra monoculaire, capable d'atteindre une telle précision pour ce type de vidéos où des objets sont fréquemment interposés entre la caméra et l'environnement, où de sévères différences d'illuminations et du flou de bougé sont présents et où l'environnement change entre les images de la base d'images et celles de la vidéo à localiser.

Conclusion et perspectives

Conclusion

Dans ce travail de thèse, nous nous sommes intéressés au problème de l'estimation de paramètres évoluant sur des groupes de Lie.

Parmi les différentes géométries que l'on pouvait choisir d'exploiter, nous avons opté pour la structure de *sous-groupe à un paramètre* du groupe de Lie. Nous avons montré que la géométrie naturelle et intuitive, induite par cette structure, permettait d'obtenir des algorithmes d'estimation à la fois efficaces et élégants.

Les algorithmes d'estimation proposés dans ce travail de thèse s'inscrivent de manière générale dans un cadre bayésien pour lequel les paramètres inconnus sont probabilisés. Pour ce faire, nous avons utilisé une généralisation possible de la distribution normale multivariée aux groupes de Lie, appelée distribution normale concentrée sur groupe de Lie. Cette distribution prend notamment en compte la géométrie du groupe de Lie de manière intrinsèque, fournissant ainsi un formalisme élégant permettant de caractériser l'incertitude des paramètres estimés.

Le filtrage de Kalman, qui est une des méthodes les plus utilisées dans le domaine de l'estimation séquentielle de paramètres euclidiens, peut être vu comme un algorithme de filtrage bayésien sous des hypothèses de bruit gaussien. Nous nous sommes par conséquent logiquement intéressés au problème du filtrage de Kalman à temps discret et continu-discret à partir d'observations appartenant à un groupe de Lie, où l'état évolue également sur un groupe de Lie. Cette étude nous a conduit à la proposition de deux filtres ; le CD-LG-EKF qui permet de résoudre un problème à temps continu-discret et le D-LG-EKF qui permet de résoudre un problème à temps discret. Ces deux filtres sont basés sur la même approximation, à savoir qu'à chaque instant, nous supposons que la distribution a posteriori de l'état est une distribution normale concentrée sur groupe de Lie. En utilisant la structure de sous-groupe à un paramètre du groupe de Lie sur lequel évoluent les paramètres, nous avons pu montrer comment propager et mettre à jour les paramètres de cette distribution, pour

finale­ment aboutir à ces deux nouveaux algorithmes.

Nous nous sommes ensuite inspirés du lien entre optimisation et filtrage de Kalman, qui a conduit au développement du filtrage Kalman étendu itéré sur espace euclidien, en le transposant à l'estimation de paramètres évoluant sur un groupe de Lie. Nous avons pu établir un lien entre l'algorithme de Gauss-Newton intrinsèque et le filtrage/lissage de Kalman sur groupe de Lie, via l'approximation de Gauss-Laplace intrinsèque. Ce lien repose, une fois de plus, sur la notion de distribution normale concentrée sur groupe de Lie puisque l'approximation de Gauss-Laplace intrinsèque est une technique permettant d'ajuster une telle distribution à une distribution a posteriori quelconque. En exploitant ce lien, nous avons montré comment obtenir une généralisation du filtre Kalman étendu itéré aux groupes de Lie, appelée LG-IEKF, ainsi qu'une généralisation du lisseur de Rauch-Tung-Striebel aux groupes de Lie, appelée LG-RTS.

Finalement, dans le but de remplacer les concepts et algorithmes d'estimation sur groupe de Lie proposés dans la thèse dans un cadre applicatif, nous avons conçu un système complet de localisation d'une caméra évoluant dans un environnement intérieur préalablement cartographié. L'approche retenue repose sur deux étapes. La première étape vise à construire une base d'images représentant l'environnement intérieur du mieux possible. La seconde étape utilise cette base d'images annotées pour localiser une nouvelle vidéo dans l'environnement.

Afin de rendre l'étape de construction de la base d'images la plus automatique possible, nous avons proposé une méthode dite de localisation et de cartographie simultanée. Cette méthode, qui repose en partie sur des approximations de Gauss-Laplace intrinsèques, est capable de cartographier un environnement intérieur à partir d'une vidéo d'apprentissage. Nous avons montré sur de nombreux jeux de données que l'approche proposée est à la fois plus robuste et plus précise que les algorithmes de l'état de l'art. De plus, pour toutes ces expériences, les paramètres de l'approche ont été fixés une fois pour toutes, prouvant sa flexibilité.

Concernant la méthode de localisation en environnement intérieur dédiée à des vidéos provenant d'une caméra monoculaire, nous avons introduit un lisseur Rao-Blackwellisé sur groupe de Lie reposant sur les algorithmes LG-IEKF et LG-RTS. Nous avons montré sur plusieurs vidéos que l'approche proposée permet d'estimer la position de la caméra avec une précision de moins d'un mètre ainsi que son orientation.

Dans chacune des étapes de construction de la base d'images puis de localisation, les concepts et algorithmes d'estimation sur groupe de Lie proposés dans la thèse, et reposant sur la géométrie induite par la structure de sous-groupe

à un paramètre, ont été utilisés à la fois pour obtenir des approches élégantes et numériquement stables mais également pour caractériser l'incertitude des paramètres estimés.

Perspectives

Dans ce travail de thèse, nous avons proposé des algorithmes d'estimation de paramètres vivant sur un groupe de Lie permettant de prendre en compte de manière intrinsèque la nature de la variété sur laquelle ils évoluent. Nous avons démontré l'intérêt de cette démarche sur certaines applications en vision par ordinateur. Ce mémoire ouvre ainsi un certain nombre de perspectives pour de futures recherches.

Parmi ces perspectives, une direction de recherche intéressante consisterait à ne pas approcher la constante de normalisation de la distribution normale concentrée sur groupe de Lie par la constante de normalisation d'une distribution normale multivariée. Cette perspective impliquerait probablement l'utilisation de techniques non plus déterministes, telles que l'approximation de Gauss-Laplace intrinsèque, mais plutôt des techniques d'approximation stochastiques, telles que l'algorithme du gradient stochastique qui a déjà été porté sur variété Riemannienne [Bonnabel 2013] ou encore l'algorithme du recuit simulé [Arnaudon 2014].

Une deuxième perspective méthodologique viserait à remplacer l'approximation de Gauss-Laplace intrinsèque, qui se base sur une technique de linéarisation, par une méthode impliquant la minimisation d'une divergence de Kullback-Leibler. Une telle approche permettrait d'obtenir une approximation à l'ordre 2 de la distribution que l'on cherche à approcher. Un travail préliminaire a déjà été mené dans cette direction, améliorant ainsi les résultats obtenus avec une technique de linéarisation [Chirikjian 2014].

Du point de vue applicatif, le système de localisation que nous avons proposé n'inclut pas de suivi de l'information contenue dans les images. De ce fait, lorsque l'algorithme de recherche par le contenu ne parvient pas à retrouver d'image correspondant à la pose courante de la caméra, l'incertitude de la pose estimée croît très rapidement. De plus, la base d'images n'est pas mise à jour avec les données issues des vidéos traitées. L'utilisation d'un algorithme de localisation et de la cartographie simultanée temps réel, tel que ceux proposés récemment dans [Engel 2014] ou [Newcombe 2011], permettrait d'atténuer ces problèmes. Les particules du lisseur Rao-Blackwellisé seraient ainsi guidées lorsque l'algorithme de recherche par le contenu rencontre des difficultés et l'environnement pourrait être mis à jour. Néanmoins, dans notre application de localisation, l'environnement est souvent dynamique et non statique, ce qui nous conduit à envisager une dernière perspective.

En effet, dans le domaine de la localisation et de la cartographie simultanée, les algorithmes de l'état de l'art, y compris celui que nous avons développé dans cette thèse, reposent en grande majorité sur une hypothèse de scène statique et cherchent à supprimer les données ne respectant pas ce modèle. Ainsi, une perspective importante consisterait à modéliser non plus une scène statique mais une scène dynamique [Tan 2013] ce qui permettrait de tirer parti de l'information dynamique présente dans les images.

Annexes

Les différentes annexes du manuscrit sont présentées ci-après.

Marginalisation de la distribution normale concentrée sur groupe de Lie

Considérons un groupe de Lie de la forme $G = G' \times G''$, où G' est un groupe de Lie matriciel de dimension p et G'' est un groupe de Lie matriciel de dimension q . Une distribution normale concentrée (à droite) peut alors s'écrire :

$$p(X) = p(X_1, X_2) = \frac{1}{\sqrt{(2\pi)^{p+q} \det(\Sigma)}} e^{-\frac{1}{2}Q(X_1, X_2)} \quad (\text{A.1})$$

où $X \in G$, $X_1 \in G'$, $X_2 \in G''$ et

$$\begin{aligned} Q(X_1, X_2) &= \left[\log_{G'}^\vee(X_1 \mu_1^{-1})^T, \log_{G''}^\vee(X_2 \mu_2^{-1})^T \right] \begin{bmatrix} \Sigma^{11} & \Sigma^{12} \\ \Sigma^{21} & \Sigma^{22} \end{bmatrix} \begin{bmatrix} \log_{G'}^\vee(X_1 \mu_1^{-1}) \\ \log_{G''}^\vee(X_2 \mu_2^{-1}) \end{bmatrix} \\ &= \log_{G'}^\vee(X_1 \mu_1^{-1})^T \Sigma^{11} \log_{G'}^\vee(X_1 \mu_1^{-1}) + \log_{G''}^\vee(X_2 \mu_2^{-1})^T \Sigma^{22} \log_{G''}^\vee(X_2 \mu_2^{-1}) \\ &\quad + 2 \log_{G'}^\vee(X_1 \mu_1^{-1})^T \Sigma^{12} \log_{G''}^\vee(X_2 \mu_2^{-1}) \end{aligned} \quad (\text{A.2})$$

avec

$$\Sigma^{-1} = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}^{-1} = \begin{bmatrix} \Sigma^{11} & \Sigma^{12} \\ \Sigma^{21} & \Sigma^{22} \end{bmatrix}$$

En utilisant les lemmes d'inversion matricielle nous avons alors :

$$\Sigma^{11} = \left(\Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{12}^T \right)^{-1} = \Sigma_{11}^{-1} + \Sigma_{11}^{-1} \Sigma_{12} \left(\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12} \right)^{-1} \Sigma_{12}^T \Sigma_{11}^{-1} \quad (\text{A.3})$$

$$\Sigma^{12} = -\Sigma_{11}^{-1} \Sigma_{12} \left(\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12} \right)^{-1} \quad (\text{A.4})$$

$$\Sigma^{22} = \left(\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12} \right)^{-1} = \Sigma_{22}^{-1} + \Sigma_{22}^{-1} \Sigma_{12}^T \left(\Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{12}^T \right)^{-1} \Sigma_{12} \Sigma_{22}^{-1} \quad (\text{A.5})$$

Après substitution, et en définissant $x_1 = \log_{G'}^\vee (X_1 \mu_1^{-1})$ et $x_2 = \log_{G''}^\vee (X_2 \mu_2^{-1})$ nous obtenons :

$$\begin{aligned}
 Q(x_1, x_2) &= x_1^T \left(\Sigma_{11}^{-1} + \Sigma_{11}^{-1} \Sigma_{12} \left(\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12} \right)^{-1} \Sigma_{12}^T \Sigma_{11}^{-1} \right) x_1 \\
 &\quad + x_2^T \left(\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12} \right)^{-1} x_2 \\
 &\quad - 2 x_1^T \Sigma_{11}^{-1} \Sigma_{12} \left(\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12} \right)^{-1} x_2 \\
 &= x_1^T \Sigma_{11}^{-1} x_1 + x_1^T \Sigma_{11}^{-1} \Sigma_{12} \left(\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12} \right)^{-1} \Sigma_{12}^T \Sigma_{11}^{-1} x_1 \\
 &\quad + x_2^T \left(\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12} \right)^{-1} x_2 \\
 &\quad - 2 x_1^T \Sigma_{11}^{-1} \Sigma_{12} \left(\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12} \right)^{-1} x_2 \\
 &= x_1^T \Sigma_{11}^{-1} x_1 + \left(\Sigma_{12}^T \Sigma_{11}^{-1} x_1 \right)^T \left(\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12} \right)^{-1} \left(\Sigma_{12}^T \Sigma_{11}^{-1} x_1 \right) \\
 &\quad + x_2^T \left(\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12} \right)^{-1} x_2 \\
 &\quad - 2 \left(\Sigma_{12}^T \Sigma_{11}^{-1} x_1 \right)^T \left(\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12} \right)^{-1} x_2 \\
 &= x_1^T \Sigma_{11}^{-1} x_1 + \left(x_2 - \Sigma_{12}^T \Sigma_{11}^{-1} x_1 \right)^T \left(\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12} \right)^{-1} \left(x_2 - \Sigma_{12}^T \Sigma_{11}^{-1} x_1 \right) \\
 &= Q_1(x_1) + Q_2(x_1, x_2) \tag{A.6}
 \end{aligned}$$

où

$$Q_1(X_1) = \log_{G'}^\vee (X_1 \mu_1^{-1})^T \Sigma_{11}^{-1} \log_{G'}^\vee (X_1 \mu_1^{-1}) \tag{A.7}$$

et

$$\begin{aligned}
 Q_2(X_1, X_2) &= \left(\log_{G''}^\vee (X_2 \mu_2^{-1}) - \Sigma_{12}^T \Sigma_{11}^{-1} \log_{G'}^\vee (X_1 \mu_1^{-1}) \right)^T \left(\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12} \right)^{-1} \\
 &\quad \left(\log_{G''}^\vee (X_2 \mu_2^{-1}) - \Sigma_{12}^T \Sigma_{11}^{-1} \log_{G'}^\vee (X_1 \mu_1^{-1}) \right) \tag{A.8}
 \end{aligned}$$

Puisque la covariance de la distribution est supposée concentrée, nous pouvons faire l'approximation suivante :

$$\log_{G''}^\vee (X_2 \mu_2^{-1}) - \Sigma_{12}^T \Sigma_{11}^{-1} \log_{G'}^\vee (X_1 \mu_1^{-1}) \approx \log_{G''}^\vee \left(X_2 \left(\mu_2 \exp_{G''}^\wedge \left(\Sigma_{12}^T \Sigma_{11}^{-1} \log_{G'}^\vee (X_1 \mu_1^{-1}) \right) \right)^{-1} \right) \tag{A.9}$$

Et donc

$$\begin{aligned}
 Q_2(X_1, X_2) &\approx \log_{G''}^\vee \left(X_2 \left(\mu_2 \exp_{G''}^\wedge \left(\Sigma_{12}^T \Sigma_{11}^{-1} \log_{G'}^\vee (X_1 \mu_1^{-1}) \right) \right)^{-1} \right)^T \left(\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12} \right)^{-1} \\
 &\quad \log_{G''}^\vee \left(X_2 \left(\mu_2 \exp_{G''}^\wedge \left(\Sigma_{12}^T \Sigma_{11}^{-1} \log_{G'}^\vee (X_1 \mu_1^{-1}) \right) \right)^{-1} \right) \tag{A.10}
 \end{aligned}$$

Ainsi nous obtenons la marginalisation suivante :

$$\begin{aligned}
& \int f(X_1, X_2) d_H X_2 \\
&= \int \frac{1}{\sqrt{(2\pi)^{p+p} \det(\Sigma_{11}) \det(\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12})}} e^{-\frac{1}{2}(Q_1(X_1) + Q_2(X_1, X_2))} d_H X_2 \\
&= \frac{1}{\sqrt{(2\pi)^p \det(\Sigma_{11})}} e^{-\frac{1}{2}Q_1(X_1)} \int \frac{1}{\sqrt{(2\pi)^q \det(\Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12})}} e^{-\frac{1}{2}Q_2(X_1, X_2)} d_H X_2 \\
&= \mathcal{N}_{G'}^R(X_1; \mu_1, \Sigma_{11}) \int \mathcal{N}_{G''}^R(X_2; \mu_2 \exp^{\wedge}_{G''}(\Sigma_{12}^T \Sigma_{11}^{-1} \log^{\vee}_{G'}(X_1 \mu_1^{-1})), \Sigma_{22} - \Sigma_{12}^T \Sigma_{11}^{-1} \Sigma_{12}) d_H X_2 \\
&= \mathcal{N}_{G'}^R(X_1; \mu_1, \Sigma_{11}) \tag{A.11}
\end{aligned}$$

Dérivées première et seconde de l'application logarithme de matrice

Nous résumons ici les approximations qui nous permettent d'établir les expressions des dérivées première et seconde de l'application logarithme de matrice dans la partie 1.1.5. Dans la suite, $Y \in G$ et $\mathfrak{b}, \mathfrak{c} \in \mathfrak{g}$.

B.1 Dérivée première de l'application logarithme de matrice : résultats intermédiaires

$$Y = \exp_G(\mathfrak{b}) = Id + \mathfrak{b} + \frac{1}{2}\mathfrak{b}^2 + O(\|\mathfrak{b}\|^3) \quad (\text{B.1})$$

$$Y\mathfrak{c} = \left(Id + \mathfrak{b} + \frac{1}{2}\mathfrak{b}^2\right)\mathfrak{c} + O(\|\mathfrak{b}\|^3)\mathfrak{c} \quad (\text{B.2})$$

$$\begin{aligned} (Y - Id)Y\mathfrak{c} &= \left(\mathfrak{b} + \frac{1}{2}\mathfrak{b}^2 + O(\|\mathfrak{b}\|^3)\right)\left(Id + \mathfrak{b} + \frac{1}{2}\mathfrak{b}^2 + O(\|\mathfrak{b}\|^3)\right)\mathfrak{c} \\ &= \left(\mathfrak{b} + \frac{3}{2}\mathfrak{b}^2\right)\mathfrak{c} + O(\|\mathfrak{b}\|^3)\mathfrak{c} \end{aligned} \quad (\text{B.3})$$

$$\begin{aligned} Y\mathfrak{c}(Y - Id) &= \left(Id + \mathfrak{b} + \frac{1}{2}\mathfrak{b}^2 + O(\|\mathfrak{b}\|^3)\right)\mathfrak{c}\left(\mathfrak{b} + \frac{1}{2}\mathfrak{b}^2 + O(\|\mathfrak{b}\|^3)\right) \\ &= \mathfrak{c}\left(\mathfrak{b} + \frac{1}{2}\mathfrak{b}^2\right) + \mathfrak{b}\mathfrak{c}\mathfrak{b} + O(\|\mathfrak{b}\|^3)\mathfrak{c} \end{aligned} \quad (\text{B.4})$$

$$\begin{aligned} Y\mathfrak{c}(Y - Id)^2 &= Y\mathfrak{c}(Y - Id)(Y - Id) \\ &= \left(\mathfrak{c}\left(\mathfrak{b} + \frac{1}{2}\mathfrak{b}^2\right) + \mathfrak{b}\mathfrak{c}\mathfrak{b} + O(\|\mathfrak{b}\|^3)\mathfrak{c}\right)\left(\mathfrak{b} + \frac{1}{2}\mathfrak{b}^2 + O(\|\mathfrak{b}\|^3)\right) \\ &= \mathfrak{c}\mathfrak{b}^2 + O(\|\mathfrak{b}\|^3)\mathfrak{c} \end{aligned} \quad (\text{B.5})$$

$$\begin{aligned}(Y - Id) Y \mathbf{c} (Y - Id) &= \left(\left(\mathbf{b} + \frac{3}{2} \mathbf{b}^2 \right) \mathbf{c} + O(\|\mathbf{b}\|^3) \mathbf{c} \right) \left(\mathbf{b} + \frac{1}{2} \mathbf{b}^2 + O(\|\mathbf{b}\|^3) \right) \\ &= \mathbf{b} \mathbf{c} \mathbf{b} + O(\|\mathbf{b}\|^3) \mathbf{c}\end{aligned}\quad (\text{B.6})$$

$$\begin{aligned}(Y - Id)^2 Y \mathbf{c} &= \left(\mathbf{b} + \frac{1}{2} \mathbf{b}^2 + O(\|\mathbf{b}\|^3) \right) \left(\left(\mathbf{b} + \frac{3}{2} \mathbf{b}^2 \right) \mathbf{c} + O(\|\mathbf{b}\|^3) \mathbf{c} \right) \\ &= \mathbf{b}^2 \mathbf{c} + O(\|\mathbf{b}\|^3) \mathbf{c}\end{aligned}\quad (\text{B.7})$$

$$\begin{aligned}O(\|(Y - Id)\|^3) Y \mathbf{c} &= O\left(\left\| \mathbf{b} + \frac{1}{2} \mathbf{b}^2 + O(\|\mathbf{b}\|^3) \right\|^3\right) \left(Id + \mathbf{b} + \frac{1}{2} \mathbf{b}^2 + O(\|\mathbf{b}\|^3) \right) \mathbf{c} \\ &= O(\|\mathbf{b}\|^3) \left(Id + \mathbf{b} + \frac{1}{2} \mathbf{b}^2 + O(\|\mathbf{b}\|^3) \right) \mathbf{c} \\ &= O(\|\mathbf{b}\|^3) \mathbf{c}\end{aligned}\quad (\text{B.8})$$

B.2 Dérivée seconde de l'application logarithme de matrice : résultats intermédiaires

$$\begin{aligned}Y \mathbf{c}^2 &= \left(Id + \mathbf{b} + \frac{1}{2} \mathbf{b}^2 \right) \mathbf{c}^2 + O(\|\mathbf{b}\|^3) \mathbf{c}^2 \\ &= (Id + \mathbf{b}) \mathbf{c}^2 + O(\|\mathbf{b}\|^2) (\mathbf{c}, \mathbf{c})\end{aligned}\quad (\text{B.9})$$

$$\begin{aligned}(Y - Id) Y \mathbf{c}^2 &= \left(\mathbf{b} + \frac{1}{2} \mathbf{b}^2 + O(\|\mathbf{b}\|^3) \right) \left((Id + \mathbf{b}) \mathbf{c}^2 + O(\|\mathbf{b}\|^2) (\mathbf{c}, \mathbf{c}) \right) \\ &= \mathbf{b} \mathbf{c}^2 + O(\|\mathbf{b}\|^2) (\mathbf{c}, \mathbf{c})\end{aligned}\quad (\text{B.10})$$

$$\begin{aligned}Y \mathbf{c}^2 (Y - Id) &= \left((Id + \mathbf{b}) \mathbf{c}^2 + O(\|\mathbf{b}\|^2) (\mathbf{c}, \mathbf{c}) \right) \left(\mathbf{b} + \frac{1}{2} \mathbf{b}^2 + O(\|\mathbf{b}\|^3) \right) \\ &= \mathbf{c}^2 \mathbf{b} + O(\|\mathbf{b}\|^2) (\mathbf{c}, \mathbf{c})\end{aligned}\quad (\text{B.11})$$

$$\begin{aligned}(Y \mathbf{c})^2 &= \left(Id + \mathbf{b} + \frac{1}{2} \mathbf{b}^2 + O(\|\mathbf{b}\|^3) \right) \mathbf{c} \left(Id + \mathbf{b} + \frac{1}{2} \mathbf{b}^2 + O(\|\mathbf{b}\|^3) \right) \mathbf{c} \\ &= \mathbf{c}^2 + \mathbf{c} \mathbf{b} \mathbf{c} + \mathbf{b} \mathbf{c}^2 + O(\|\mathbf{b}\|^2) (\mathbf{c}, \mathbf{c})\end{aligned}\quad (\text{B.12})$$

$$\begin{aligned} (Y\mathbf{c})^2 (Y - Id) &= \left(\mathbf{c}^2 + \mathbf{c}\mathbf{b}\mathbf{c} + \mathbf{b}\mathbf{c}^2 \right) \left(\mathbf{b} + \frac{1}{2}\mathbf{b}^2 \right) + O\left(\|\mathbf{b}\|^2\right) (\mathbf{c}, \mathbf{c}) \\ &= \mathbf{c}^2\mathbf{b} + O\left(\|\mathbf{b}\|^2\right) (\mathbf{c}, \mathbf{c}) \end{aligned} \quad (\text{B.13})$$

$$\begin{aligned} Y\mathbf{c} (Y - Id) Y\mathbf{c} &= \left(Id + \mathbf{b} + \frac{1}{2}\mathbf{b}^2 + O\left(\|\mathbf{b}\|^3\right) \right) \mathbf{c} \left(\left(\mathbf{b} + \frac{3}{2}\mathbf{b}^2 \right) \mathbf{c} + O\left(\|\mathbf{b}\|^3\right) \mathbf{c} \right) \\ &= \mathbf{c}\mathbf{b}\mathbf{c} + O\left(\|\mathbf{b}\|^2\right) (\mathbf{c}, \mathbf{c}) \end{aligned} \quad (\text{B.14})$$

$$\begin{aligned} (Y - Id) (Y\mathbf{c})^2 &= \left(\mathbf{b} + \frac{1}{2}\mathbf{b}^2 \right) \left(\mathbf{c}^2 + \mathbf{c}\mathbf{b}\mathbf{c} + \mathbf{b}\mathbf{c}^2 \right) + O\left(\|\mathbf{b}\|^2\right) (\mathbf{c}, \mathbf{c}) \\ &= \mathbf{b}\mathbf{c}^2 + O\left(\|\mathbf{b}\|^2\right) (\mathbf{c}, \mathbf{c}) \end{aligned} \quad (\text{B.15})$$

$$O\left(\|(Y - Id)\|^2\right) (Y\mathbf{c}, Y\mathbf{c}) = O\left(\|(Y - Id)\|^2\right) (\mathbf{c}, \mathbf{c}) \quad (\text{B.16})$$

$$\begin{aligned} O\left(\|(Y - Id)\|^3\right) Y\mathbf{c}^2 &= O\left(\|(Y - Id)\|^3\right) (Y\mathbf{c}, Y\mathbf{c}) \\ &= O\left(\|(Y - Id)\|^2\right) (\mathbf{c}, \mathbf{c}) \end{aligned} \quad (\text{B.17})$$

$$\begin{aligned} O\left(\|Y - Id\|^2\right) (\mathbf{c}, \mathbf{c}) &= O\left(\left\| \mathbf{b} + \frac{1}{2}\mathbf{b}^2 + O\left(\|\mathbf{b}\|^3\right) \right\|^2\right) (\mathbf{c}, \mathbf{c}) \\ &= O\left(\|\mathbf{b}\|^2\right) (\mathbf{c}, \mathbf{c}) \end{aligned} \quad (\text{B.18})$$

Propriétés de l'opérateur $\text{ad}_G(\cdot)$

Nous nous intéressons dans cette annexe à certaines propriétés de l'opérateur $\text{ad}_G(\cdot)$ que nous utilisons dans le chapitre 2.

En utilisant les définitions :

$$(\text{ad}_G(x))_{ij} = L_{ij}^T x \quad (\text{C.1})$$

et

$$(\text{ad}_G(x)^T)_{ij} = (\text{ad}_G(x))_{ji} = L_{ji}^T x \quad (\text{C.2})$$

où $L_{ij} \in \mathbb{R}^p$ et $x \in \mathbb{R}^p$, nous obtenons alors :

$$\begin{aligned} & (\text{ad}_G(x) \text{Rad}_G(x)^T)_{ij} \\ &= \mathbb{E} \left(\sum_{k=1}^p \sum_{l=1}^p (\text{ad}_G(x))_{ik} R_{kl} (\text{ad}_G(x)^T)_{lj} \right) \\ &= \mathbb{E} \left(\sum_{k=1}^p \sum_{l=1}^p L_{ik}^T x R_{kl} L_{jl}^T x \right) \\ &= \mathbb{E} \left(\sum_{k=1}^p \sum_{l=1}^p R_{kl} L_{ik}^T x x^T L_{jl} \right) \\ &= \sum_{k=1}^p \sum_{l=1}^p R_{kl} L_{ik}^T x x^T L_{jl} \end{aligned} \quad (\text{C.3})$$

et

$$\begin{aligned}
 \left(\text{ad}_G(x)^2\right)_{ij} &= \sum_{k=1}^p (\text{ad}_G(x))_{ik} (\text{ad}_G(x))_{kj} \\
 &= \sum_{k=1}^p L_{ik}^T x L_{kj}^T x \\
 &= \sum_{k=1}^p L_{ik}^T x x^T L_{kj}
 \end{aligned} \tag{C.4}$$

Moyenne et covariance d'un système dynamique

Nous présentons dans cette annexe les équations de propagation de la moyenne et de la covariance d'un système dynamique décrit par l'équation différentielle stochastique suivante :

$$dx = f(x) dt + g(x) d\beta \quad (\text{D.1})$$

où f et g sont des fonctions bornées et lipschitziennes (voir [Maybeck 1979], Volume 2, pp.182), β est un mouvement brownien de matrice de diffusion $Q(t)$, et $x \in \mathbb{R}^m$.

Les équations de propagations de la moyenne et de la covariance sont (voir [Maybeck 1979] Volume 2, Chapitre 11, Section 6) :

$$\dot{m}_x = \mathbb{E}[f(x)] \quad (\text{D.2})$$

$$\begin{aligned} \dot{P}_x = & \left(\mathbb{E}[f(x) x^T] - \mathbb{E}[f(x)] m_x^T \right) \\ & + \left(\mathbb{E}[x f(x)^T] - m_x \mathbb{E}[f(x)^T] \right) \\ & + \mathbb{E}[g(x) Q g(x)^T] \end{aligned} \quad (\text{D.3})$$

Complétion du carré

Nous présentons dans cette annexe la méthode dite de “complétion du carré”.
Considérons la norme de Mahalanobis suivante :

$$\begin{aligned}\|Ax - b\|_{\Sigma}^2 &= (Ax - b)^T \Sigma^{-1} (Ax - b) \\ &= x^T A^T \Sigma^{-1} Ax - x^T A^T \Sigma^{-1} b - b^T \Sigma^{-1} Ax + cst_1\end{aligned}\quad (\text{E.1})$$

où cst_1 est un terme constant par rapport à x . En ajoutant le terme $(A^T \Sigma^{-1} b)^T (A^T \Sigma^{-1} A) (A^T \Sigma^{-1} b)$, qui est constant par rapport à x , nous obtenons :

$$\begin{aligned}\|Ax - b\|_{\Sigma}^2 &= x^T A^T \Sigma^{-1} Ax - x^T A^T \Sigma^{-1} b - b^T \Sigma^{-1} Ax \\ &\quad + (A^T \Sigma^{-1} b)^T (A^T \Sigma^{-1} A) (A^T \Sigma^{-1} b) + cst_2\end{aligned}\quad (\text{E.2})$$

Nous définissons $P^{-1} = A^T \Sigma^{-1} A$ et injectons PP^{-1} ou $P^{-1}P$ dans chaque terme :

$$\begin{aligned}\|Ax - b\|_{\Sigma}^2 &= x^T P^{-1} x - x^T P^{-1} P A^T \Sigma^{-1} b - b^T \Sigma^{-1} A P P^{-1} x \\ &\quad + (A^T \Sigma^{-1} b)^T P P^{-1} P (A^T \Sigma^{-1} b) + cst_2\end{aligned}\quad (\text{E.3})$$

Remarquons que le minimiseur de $\|Ax - b\|_{\Sigma}^2$ est :

$$\begin{aligned}\hat{x} &= (A^T \Sigma^{-1} A)^{-1} A^T \Sigma^{-1} b \\ &= P A^T \Sigma^{-1} b\end{aligned}\quad (\text{E.4})$$

Nous obtenons finalement :

$$\begin{aligned}\|Ax - b\|_{\Sigma}^2 &= x^T P^{-1} x - x^T P^{-1} \hat{x} - \hat{x}^T P^{-1} x + \hat{x}^T P^{-1} \hat{x} + cst_2 \\ &= (x - \hat{x})^T P^{-1} (x - \hat{x}) + cst_2\end{aligned}\tag{E.5}$$

Ainsi, nous avons “complété le carré”, c’est à dire que nous avons fait apparaître la distance de Mahalanobis entre x et \hat{x} .

Lemmes d'inversion matricielle

Nous présentons dans cette annexe les différents lemmes d'inversion matricielle dont nous faisons usage dans le mémoire.

F.1 Lemme 1

$$\begin{bmatrix} A & B \\ B^T & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}BE^{-1}B^TA^{-1} & -A^{-1}BE^{-1} \\ -E^{-1}B^TA^{-1} & E^{-1} \end{bmatrix} \quad (\text{F.1})$$

où

$$E = D - B^TA^{-1}B \quad (\text{F.2})$$

F.2 Lemme 2

$$\begin{bmatrix} A & B \\ B^T & D \end{bmatrix}^{-1} = \begin{bmatrix} G^{-1} & -G^{-1}BD^{-1} \\ -D^{-1}B^TG^{-1} & D^{-1} + D^{-1}B^TG^{-1}BD^{-1} \end{bmatrix} \quad (\text{F.3})$$

où

$$G = A - BD^{-1}B^T \quad (\text{F.4})$$

F.3 Lemme 3

$$(X + V^T W V)^{-1} = X^{-1} - X^{-1} V^T (W^{-1} + V X^{-1} V^T)^{-1} V X^{-1} \quad (\text{F.5})$$

F.4 Lemme 4

$$(X + Y)^{-1} (X + Y) = Id$$

donc

$$Id - (X + Y)^{-1} X = (X + Y)^{-1} Y$$

d'où

$$\left(Id - (X + Y)^{-1} X \right) Y^{-1} = (X + Y)^{-1} \quad (\text{F.6})$$

Démonstration du lisseur de Rauch–Tung–Striebel sur groupe de Lie

Les différents résultats en lien avec la démonstration du lisseur de Rauch–Tung–Striebel sur groupe de Lie (voir chapitre 3) sont décrits dans cette annexe.

G.1 Résultat 1

Posons :

$$L_k = P_{k|k} F_k^T P_{k+1|k}^{-1} = P_{k|k} F_k^T \left(R_k + F_k P_{k|k} F_k^T \right)^{-1} \quad (\text{G.1})$$

et

$$P_{k+1|k} L_k^T = P_{k+1|k} P_{k+1|k}^{-1} F_k P_{k|k} = F_k P_{k|k} \quad (\text{G.2})$$

En utilisant (F.5) puis (G.1) et (G.2) on démontre :

$$\left(F_k^T R_k^{-1} F_k + P_{k|k}^{-1} \right)^{-1} = P_{k|k} - P_{k|k} F_k^T \left(R_k + F_k P_{k|k} F_k^T \right)^{-1} F_k P_{k|k} \quad (\text{G.3})$$

$$\begin{aligned} &= P_{k|k} - L_k F_k P_{k|k} \\ &= P_{k|k} - L_k P_{k+1|k} L_k^T \end{aligned} \quad (\text{G.4})$$

G.2 Résultat 2

En utilisant l'égalité suivante qui se base sur (F.6) :

$$\begin{aligned} &\left(Id - P_{k+1|k}^{-1} F_k P_{k|k} F_k^T \right) R_k^{-1} \\ &= \left(Id - \left(R_k + F_k P_{k|k} F_k^T \right)^{-1} F_k P_{k|k} F_k^T \right) R_k^{-1} \\ &= \left(R_k + F_k P_{k|k} F_k^T \right)^{-1} = P_{k+1|k}^{-1} \end{aligned} \quad (\text{G.5})$$

On démontre que :

$$\begin{aligned}
 D^{-1}B^T &= \mathcal{S}^{-1} \left(P_{k|k} - P_{k|k} F_k^T P_{k+1|k}^{-1} F_k P_{k|k} \right) \mathcal{S}^{-T} \left(-\mathcal{S}^T F_k^T R_k^{-1} \mathcal{M} \right) \\
 &= -\mathcal{S}^{-1} \left(P_{k|k} - P_{k|k} F_k^T P_{k+1|k}^{-1} F_k P_{k|k} \right) \left(F_k^T R_k^{-1} \mathcal{M} \right) \\
 &= -\mathcal{S}^{-1} P_{k|k} F_k^T \left(Id - P_{k+1|k}^{-1} F_k P_{k|k} F_k^T \right) R_k^{-1} \mathcal{M} \\
 &= -\mathcal{S}^{-1} P_{k|k} F_k^T P_{k+1|k}^{-1} \mathcal{M} \\
 &= -\mathcal{S}^{-1} L_k \mathcal{M}
 \end{aligned} \tag{G.6}$$

ce qui prouve au passage que

$$\left(P_{k|k} - P_{k|k} F_k^T P_{k+1|k}^{-1} F_k P_{k|k} \right) F_k^T R_k^{-1} = L_k \tag{G.7}$$

G.3 Résultat 3

D'après (F.5)

$$P_{k+1|k}^{-1} = \left(R_k + F_k P_{k|k} F_k^T \right)^{-1} = R_k^{-1} - R_k^{-1} F_k \left(P_{k|k}^{-1} + F_k^T R_k^{-1} F_k \right)^{-1} F_k^T R_k^{-1} \tag{G.8}$$

G.4 Résultat 4

D'après (F.6) :

$$\begin{aligned}
 P_{k+1|k}^{-1} &= \left(R_k + F_k P_{k|k} F_k^T \right)^{-1} \\
 &= \left(Id - \left(R_k + F_k P_{k|k} F_k^T \right)^{-1} F_k P_{k|k} F_k^T \right) R_k^{-1} \\
 &= \left(Id - P_{k+1|k}^{-1} F_k P_{k|k} F_k^T \right) R_k^{-1}
 \end{aligned} \tag{G.9}$$

donc

$$R_k^{-1} - P_{k+1|k}^{-1} = R_k^{-1} - \left(Id - P_{k+1|k}^{-1} F_k P_{k|k} F_k^T \right) R_k^{-1} \tag{G.10}$$

$$= P_{k+1|k}^{-1} F_k P_{k|k} F_k^T R_k^{-1} \tag{G.11}$$

Finalement en utilisant (G.1), on a :

$$\begin{aligned}
(R_k^{-1} - P_{k+1|k}^{-1}) - L_k^T F_k^T R_k^{-1} &= R_k^{-1} - P_{k+1|k}^{-1} - P_{k+1|k}^{-1} F_k P_{k|k} F_k^T R_k^{-1} \\
&= P_{k+1|k}^{-1} F_k P_{k|k} F_k^T R_k^{-1} - P_{k+1|k}^{-1} F_k P_{k|k} F_k^T R_k^{-1} \\
&= 0
\end{aligned} \tag{G.12}$$

G.5 Résultat 5

D'après (G.8) et (G.1) :

$$\begin{aligned}
R_k^{-1} F_k - L_k^T (F_k^T R_k^{-1} F_k + P_{k|k}^{-1}) \\
&= R_k^{-1} F_k - P_{k+1|k}^{-1} F_k P_{k|k} (F_k^T R_k^{-1} F_k + P_{k|k}^{-1}) \\
&= R_k^{-1} F_k - \left(R_k^{-1} - R_k^{-1} F_k (P_{k|k}^{-1} + F_k^T R_k^{-1} F_k)^{-1} F_k^T R_k^{-1} \right) F_k P_{k|k} (F_k^T R_k^{-1} F_k + P_{k|k}^{-1})
\end{aligned} \tag{G.13}$$

Or d'après (F.6)

$$\left(Id - (P_{k|k}^{-1} + F_k^T R_k^{-1} F_k)^{-1} F_k^T R_k^{-1} F_k \right) P_{k|k} = (P_{k|k}^{-1} + F_k^T R_k^{-1} F_k)^{-1} \tag{G.14}$$

Donc

$$\begin{aligned}
&\left(R_k^{-1} - R_k^{-1} F_k (P_{k|k}^{-1} + F_k^T R_k^{-1} F_k)^{-1} F_k^T R_k^{-1} \right) F_k P_{k|k} \\
&= R_k^{-1} F_k \left(Id - (P_{k|k}^{-1} + F_k^T R_k^{-1} F_k)^{-1} F_k^T R_k^{-1} F_k \right) P_{k|k} \\
&= R_k^{-1} F_k (P_{k|k}^{-1} + F_k^T R_k^{-1} F_k)^{-1}
\end{aligned} \tag{G.15}$$

Donc en reprenant (G.13) et en utilisant (G.15), on obtient :

$$\begin{aligned}
R_k^{-1} F_k - L_k^T (F_k^T R_k^{-1} F_k + P_{k|k}^{-1}) \\
&= R_k^{-1} F_k - R_k^{-1} F_k (P_{k|k}^{-1} + F_k^T R_k^{-1} F_k)^{-1} (F_k^T R_k^{-1} F_k + P_{k|k}^{-1}) \\
&= 0
\end{aligned} \tag{G.16}$$

G.6 Calcul de $(J_l^T \mathfrak{E}^{-1} J_l)^{-1}$

Calculons l'inverse de chaque bloc de $(J_l^T \mathfrak{E}^{-1} J_l)^{-1}$:

$$(J_l^T \mathfrak{E}^{-1} J_l)^{-1} = \begin{bmatrix} \mathcal{M}^T (R_k^{-1} - P_{k+1|k}^{-1}) \mathcal{M} + \mathcal{L}^T P_{k+1|T}^{-1} \mathcal{L} & -\mathcal{M}^T R_k^{-1} F_k \mathcal{S} \\ -\mathcal{S}^T F_k^T R_k^{-1} \mathcal{M} & \mathcal{S}^T (F_k^T R_k^{-1} F_k + P_{k|k}^{-1}) \mathcal{S} \end{bmatrix}^{-1} \quad (\text{G.17})$$

$$= \begin{bmatrix} A & B \\ B^T & D \end{bmatrix}^{-1} = \begin{bmatrix} A' & B' \\ B'^T & D' \end{bmatrix} \quad (\text{G.18})$$

Calculons A' : D'après (F.3)

$$\begin{aligned} A' &= (A - BD^{-1}B^T)^{-1} \\ &= \left(\mathcal{M}^T (R_k^{-1} - P_{k+1|k}^{-1} - R_k^{-1} F_k (F_k^T R_k^{-1} F_k + P_{k|k}^{-1})^{-1} F_k^T R_k^{-1}) \mathcal{M} + \mathcal{L}^T P_{k+1|T}^{-1} \mathcal{L} \right)^{-1} \end{aligned} \quad (\text{G.19})$$

Donc d'après (G.8), nous avons :

$$A' = (\mathcal{L}^T P_{k+1|T}^{-1} \mathcal{L})^{-1} = \mathcal{L}^{-1} P_{k+1|T} \mathcal{L}^{-T} \quad (\text{G.20})$$

Calculons D' : D'après (F.3), puis en appliquant (G.4), (G.6) et (G.20), on a :

$$\begin{aligned} D' &= D^{-1} + D^{-1} B^T A' B D^{-1} \\ &= \mathcal{S}^{-1} (P_{k|k} + L_k (\mathcal{M} \mathcal{L}^{-1} P_{k+1|T} \mathcal{L}^{-T} \mathcal{M}^T - P_{k+1|k}) L_k^T) \mathcal{S}^{-T} \end{aligned} \quad (\text{G.21})$$

où

$$L_k = P_{k|k} F_k^T P_{k+1|k}^{-1} \quad (\text{G.22})$$

Calculons B' : D'après (G.20) et (G.6) :

$$B' = -G^{-1} B D^{-1} = \mathcal{L}^{-1} P_{k+1|T} \mathcal{L}^{-T} \mathcal{M}^T L_k^T \mathcal{S}^{-T} \quad (\text{G.23})$$

Finalement nous avons :

$$\begin{aligned} (J_l^T \mathfrak{E}^{-1} J_l)^{-1} &= \\ &= \begin{bmatrix} \mathcal{L}^{-1} P_{k+1|T} \mathcal{L}^{-T} & \mathcal{L}^{-1} P_{k+1|T} \mathcal{L}^{-T} \mathcal{M}^T L_k^T \mathcal{S}^{-T} \\ \mathcal{S}^{-1} L_k \mathcal{M} \mathcal{L}^{-1} P_{k+1|T} \mathcal{L}^{-T} & \mathcal{S}^{-1} (P_{k|k} + L_k (\mathcal{M} \mathcal{L}^{-1} P_{k+1|T} \mathcal{L}^{-T} \mathcal{M}^T - P_{k+1|k}) L_k^T) \mathcal{S}^{-T} \end{bmatrix} \end{aligned} \quad (\text{G.24})$$

G.7 Calcul de $\delta_k^{l+1/l}$

En utilisant (3.110), (3.113) et (3.115), et le fait que $\delta_{k+1}^{l,T} = 0$ et $\mathcal{L} = Id$, nous obtenons :

$$\begin{aligned} \delta_k^{l+1/l} = & -\mathcal{S}^{-1} \left(L_k \mathcal{M} P_{k+1|T} \mathcal{M}^T \left(R_k^{-1} - (R_k + F_k P_{k|k} F_k^T)^{-1} \right) \right. \\ & - \left(P_{k|k} + L_k \left(\mathcal{M} P_{k+1|T} \mathcal{M}^T - P_{k+1|k} \right) L_k^T \right) F_k^T R_k^{-1} \delta_{k+1}^{l,k} \\ & - \mathcal{S}^{-1} \left(-L_k \mathcal{M} P_{k+1|T} \mathcal{M}^T R_k^{-1} F_k \right. \\ & \left. + \left(P_{k|k} + L_k \left(\mathcal{M} P_{k+1|T} \mathcal{M}^T - P_{k+1|k} \right) L_k^T \right) \left(F_k^T R_k^{-1} F_k + P_{k|k}^{-1} \right) \right) \delta_k^{l,k} \end{aligned}$$

Calcul du terme en $\delta_k^{l,k}$

Calcul du terme en $L_k \mathcal{M} P_{k+1|T} \mathcal{M}^T$

$$\begin{aligned} & -L_k \mathcal{M} P_{k+1|T} \mathcal{M}^T R_k^{-1} F_k + \left(L_k \mathcal{M} P_{k+1|T} \mathcal{M}^T L_k^T \right) \left(F_k^T R_k^{-1} F_k + P_{k|k}^{-1} \right) \\ & = L_k \mathcal{M} P_{k+1|T} \mathcal{M}^T \left(L_k^T \left(F_k^T R_k^{-1} F_k + P_{k|k}^{-1} \right) - R_k^{-1} F_k \right) \end{aligned} \quad (\text{G.26})$$

donc d'après (G.16) le terme est nul.

Calcul du terme restant

$$\begin{aligned} & \left(P_{k|k} - L_k P_{k+1|k} L_k^T \right) \left(F_k^T R_k^{-1} F_k + P_{k|k}^{-1} \right) \\ & = \left(P_{k|k} - L_k \left(R_k + F_k P_{k|k} F_k^T \right) L_k^T \right) \left(F_k^T R_k^{-1} F_k + P_{k|k}^{-1} \right) \end{aligned} \quad (\text{G.27})$$

donc d'après (G.3) le terme restant vaut Id .

Le terme en $\delta_k^{l,k}$ vaut donc $-\mathcal{S}^{-1} \delta_k^{l,k}$.

Calcul du terme en $\delta_{k+1}^{l,k}$

Calcul du terme en $L_k \mathcal{M} P_{k+1|T} \mathcal{M}^T$

$$\begin{aligned} & L_k \mathcal{M} P_{k+1|T} \mathcal{M}^T \left(R_k^{-1} - (R_k + F_k P_{k|k} F_k^T)^{-1} \right) - \left(L_k \mathcal{M} P_{k+1|T} \mathcal{M}^T L_k^T \right) F_k^T R_k^{-1} \\ & = L_k \mathcal{M} P_{k+1|T} \mathcal{M}^T \left(-L_k^T F_k^T R_k^{-1} + R_k^{-1} - (R_k + F_k P_{k|k} F_k^T)^{-1} \right) \end{aligned} \quad (\text{G.28})$$

En appliquant (G.10), on obtient :

$$\begin{aligned}
 & L_k \mathcal{M} P_{k+1|T} \mathcal{M}^T \left(-L_k^T F_k^T R_k^{-1} + R_k^{-1} - \left(R_k + F_k P_{k|k} F_k^T \right)^{-1} \right) \\
 &= L_k \mathcal{M} P_{k+1|T} \mathcal{M}^T \left(-L_k^T F_k^T R_k^{-1} + \left(R_k + F_k P_{k|k} F_k^T \right)^{-1} F_k P_{k|k} F_k^T R_k^{-1} \right) \\
 &= L_k \mathcal{M} P_{k+1|T} \mathcal{M}^T \left(\left(-L_k^T + \left(R_k + F_k P_{k|k} F_k^T \right)^{-1} F_k P_{k|k} \right) F_k^T R_k^{-1} \right) \quad (\text{G.29})
 \end{aligned}$$

Et donc d'après (G.1) le terme en $L_k \mathcal{M} P_{k+1|T} \mathcal{M}^T$ vaut 0.

Calcul du terme restant En utilisant (G.7), on montre que :

$$\begin{aligned}
 & - \left(P_{k|k} - L_k P_{k+1|k} L_k^T \right) F_k^T R_k^{-1} \\
 &= - \left(P_{k|k} - P_{k|k} F_k^T P_{k+1|k}^{-1} P_{k+1|k} P_{k+1|k}^{-1} F_k P_{k|k} \right) F_k^T R_k^{-1} \\
 &= - \left(P_{k|k} - L_k F_k P_{k|k} \right) F_k^T R_k^{-1} \\
 &= -L_k \quad (\text{G.30})
 \end{aligned}$$

Le terme en $\delta_{k+1}^{l,k}$ vaut donc $\mathcal{S}^{-1} L_k \delta_{k+1}^{l,k}$.

Au final, on a donc

$$\delta_k^{l+1/l} = \mathcal{S}^{-1} L_k \delta_{k+1}^{l,k} \quad (\text{G.31})$$

Démonstration du lisseur particulaire Rao-Blackwellisé sur groupe de Lie

Nous présentons ici les calculs permettant d'aboutir à l'algorithme du lisseur particulaire Rao-Blackwellisé sur groupe de Lie. Cette démonstration est inspirée de celle proposée dans [Fong 2002]. A la différence que la partie de l'état marginalisée évolue sur un groupe de Lie.

H.1 Étape de filtrage

Nous souhaitons approcher la distribution de probabilité suivante :

$$p(X_{0:k}, s_{0:k} | Z_{1:k}) = p(X_{0:k} | Z_{1:k}, s_{0:k}) p(s_{0:k} | Z_{1:k}) \quad (\text{H.1})$$

Le terme $p(s_{0:k} | Z_{1:k})$ sera échantillonné par N_p particules de la manière suivante (tirées selon une distribution de proposition $q(s_{0:k} | Z_{1:k})$) :

$$p(s_{0:k} | Z_{1:k}) \approx \sum_{l=1}^{N_p} w_k^{(l)} \delta(s_{0:k} - s_{0:k}^{(l)}) \quad (\text{H.2})$$

où

$$w_k^{(l)} \propto \frac{p(s_{0:k}^{(l)} | Z_{1:k})}{q(s_{0:k}^{(l)} | Z_{1:k})} \quad (\text{H.3})$$

Le terme $p(X_{0:k} | Z_{1:k}, s_{0:k})$ sera approché par une distribution normale concentrée sur groupe de Lie en utilisant le filtre de Kalman étendu sur groupe de Lie (LG-EKF) présenté dans le chapitre 3. L'efficacité de cet algorithme réside dans le fait que seul $s_{0:k}$ est échantillonné, c'est à dire que $X_{0:k}$ a été marginalisé dans (H.2). Sachant $s_{0:k}$, la partie $X_{0:k}$ de l'état marginalisée peut alors être estimée analytiquement (LG-EKF).

H.1.1 Formulation récursive

Pour garantir une formulation récursive de l'algorithme, la distribution de proposition doit se décomposer de la manière suivante :

$$q(s_{0:k}|Z_{1:k}) = q(s_k|s_{0:k-1}, Z_{1:k}) q(s_{0:k-1}|Z_{1:k-1}) \quad (\text{H.4})$$

Les particules peuvent alors être échantillonnées séquentiellement :

$$s_{0:k}^{(l)} = [s_k^{(l)}, s_{0:k-1}^{(l)}] \quad (\text{H.5})$$

avec

$$s_k^{(l)} \sim q(s_k^{(l)}|s_{0:k-1}^{(l)}, Z_{1:k}) \quad (\text{H.6})$$

De plus, la distribution cible qui nous intéresse peut être exprimée de la manière suivante :

$$\begin{aligned} p(s_{0:k}^{(l)}|Z_{1:k}) &= \frac{p(Z_k|s_{0:k}^{(l)}, Z_{1:k-1}) p(s_{0:k}^{(l)}|Z_{1:k-1})}{p(Z_k|Z_{1:k-1})} \\ &= \frac{p(Z_k|s_{0:k}^{(l)}, Z_{1:k-1}) p(s_k^{(l)}|s_{0:k-1}^{(l)}, Z_{1:k-1}) p(s_{0:k-1}^{(l)}|Z_{1:k-1})}{p(Z_k|Z_{1:k-1})} \end{aligned} \quad (\text{H.7})$$

Ainsi, nous obtenons la formule récursive suivante pour les poids $w_k^{(l)}$:

$$\begin{aligned} w_k^{(l)} &\propto \frac{p(s_{0:k}^{(l)}|Z_{1:k})}{q(s_{0:k}^{(l)}|Z_{1:k})} \propto \frac{p(Z_k|s_{0:k}^{(l)}, Z_{1:k-1}) p(s_{0:k}^{(l)}|Z_{1:k-1}) p(s_{0:k-1}^{(l)}|Z_{1:k-1})}{p(Z_k|Z_{1:k-1}) q(s_k^{(l)}|s_{0:k-1}^{(l)}, Z_{1:k-1}) q(s_{0:k-1}^{(l)}|Z_{1:k-1})} \\ &\propto w_{k-1}^{(l)} \frac{p(Z_k|s_{0:k}^{(l)}, Z_{1:k-1}) p(s_k^{(l)}|s_{0:k-1}^{(l)}, Z_{1:k-1})}{p(Z_k|Z_{1:k-1}) q(s_k^{(l)}|s_{0:k-1}^{(l)}, Z_{1:k-1})} \end{aligned} \quad (\text{H.8})$$

H.1.2 Échantillonnage optimal

H.1.2.1 Simulation de s_k

Une manière de réduire le nombre de particules est d'échantillonner les s_k selon la loi de proposition dite "optimale". C'est à dire :

$$q(s_k^{(l)}|s_{0:k-1}^{(l)}, Z_{1:k}) = p(s_k^{(l)}|s_{0:k-1}^{(l)}, Z_{1:k}) = \frac{p(Z_k|s_{0:k}^{(l)}, Z_{1:k-1}) p(s_k^{(l)}|s_{0:k-1}^{(l)}, Z_{1:k-1})}{p(Z_k|s_{0:k-1}^{(l)}, Z_{1:k-1})} \quad (\text{H.9})$$

$$\propto p(Z_k|s_{0:k}^{(l)}, Z_{1:k-1}) p(s_k^{(l)}|s_{0:k-1}^{(l)}, Z_{1:k-1}) \quad (\text{H.10})$$

Concernant le second terme de l'équation précédente, dans notre application, nous avons l'indépendance conditionnelle suivante :

$$p\left(s_k^{(l)} | s_{0:k-1}^{(l)}, Z_{1:k-1}\right) = p\left(s_k^{(l)}\right) \quad (\text{H.11})$$

Concernant le premier terme, nous avons :

$$\begin{aligned} & p\left(Z_k | s_k^{(l)}, s_{0:k-1}^{(l)}, Z_{1:k-1}\right) \\ &= \int p\left(X_k, Z_k | s_k^{(l)}, s_{0:k-1}^{(l)}, Z_{1:k-1}\right) d_H X_k \\ &= \int p\left(Z_k | X_k, s_k^{(l)}\right) p\left(X_k | s_k^{(l)}, s_{0:k-1}^{(l)}, Z_{1:k-1}\right) d_H X_k \end{aligned} \quad (\text{H.12})$$

Or d'après le modèle d'observation (4.16), nous avons :

$$p\left(Z_k | X_k, s_k^{(l)}\right) \propto \mathcal{N}_{SE(3)}^R\left(Z_k\left(s_k^{(l)}\right); C_k, Q_{k, s_k^{(l)}}\right) \quad (\text{H.13})$$

De plus, le terme $p\left(X_k | s_k^{(l)}, s_{0:k-1}^{(l)}, Z_{1:k-1}\right)$ peut être approché par une distribution normale concentrée sur groupe de Lie en utilisant l'algorithme LG-EKF :

$$p\left(X_k | s_k^{(l)}, s_{0:k-1}^{(l)}, Z_{1:k-1}\right) = \mathcal{N}_G^R\left(X_k; \mu_{k|k-1}^{(l)}, P_{k|k-1}^{(l)}\right) \quad (\text{H.14})$$

Nous avons donc :

$$\begin{aligned} & p\left(Z_k | s_k^{(l)}, s_{0:k-1}^{(l)}, Z_{1:k-1}\right) \\ & \propto \int \mathcal{N}_{SE(3)}^R\left(Z_k\left(s_k^{(l)}\right); C_k, Q_{k, s_k^{(l)}}\right) \mathcal{N}_G^R\left(X_k; \mu_{k|k-1}^{(l)}, P_{k|k-1}^{(l)}\right) d_H X_k \end{aligned} \quad (\text{H.15})$$

En appliquant une approximation de Gauss-Laplace à l'intégrande de (H.15) suivi d'une marginalisation de X_k , nous obtenons :

$$p\left(Z_k | s_k^{(l)}, s_{0:k-1}^{(l)}, Z_{1:k-1}\right) \propto \mathcal{N}_{SE(3)}^R\left(Z_k\left(s_k^{(l)}\right); \mu_{k|k-1}^{(l)}(C), P_{k|k-1}^{(l)}(C) + Q_{k, s_k^{(l)}}\right) \quad (\text{H.16})$$

Les notations $\mu_{k|k-1}^{(l)}(C)$ et $P_{k|k-1}^{(l)}(C)$ correspondent à extraire la sous-matrice correspondant à la pose de la caméra C dans $\mu_{k|k-1}^{(l)}$ et $P_{k|k-1}^{(l)}$. Ainsi nous avons obtenu le résultat suivant :

$$q\left(s_k^{(l)} | s_{0:k-1}^{(l)}, Z_{1:k}\right) \propto \mathcal{N}_{SE(3)}^R\left(Z_k\left(s_k^{(l)}\right); \mu_{k|k-1}^{(l)}(C), P_{k|k-1}^{(l)}(C) + Q_{k, s_k^{(l)}}\right) p\left(s_k^{(l)}\right) \quad (\text{H.17})$$

H.1.2.2 Calcul du poids

Nous obtenons alors la formule suivante pour les poids $w_k^{(l)}$:

$$\begin{aligned} w_k^{(l)} &\propto w_{k-1}^{(l)} \frac{p(Z_k | s_{0:k}^{(l)}, Z_{1:k-1}) p(s_k^{(l)} | s_{0:k-1}^{(l)}, Z_{1:k-1})}{p(Z_k | Z_{1:k-1})} \cdot \frac{p(Z_k | s_{0:k-1}^{(l)}, Z_{1:k-1})}{p(Z_k | s_{0:k}^{(l)}, Z_{1:k-1}) p(s_k^{(l)} | s_{0:k-1}^{(l)}, Z_{1:k-1})} \\ &\propto w_{k-1}^{(l)} \frac{p(Z_k | s_{0:k-1}^{(l)}, Z_{1:k-1})}{p(Z_k | Z_{1:k-1})} \propto w_{k-1}^{(l)} p(Z_k | s_{0:k-1}^{(l)}, Z_{1:k-1}) \end{aligned} \quad (\text{H.18})$$

Le terme $p(Z_k | s_{0:k-1}^{(l)}, Z_{1:k-1})$ peut être calculé de la manière suivante :

$$\begin{aligned} p(Z_k | s_{0:k-1}^{(l)}, Z_{1:k-1}) &= \sum_{s_k} p(Z_k, s_k | s_{0:k-1}^{(l)}, Z_{1:k-1}) \\ &= \sum_{s_k} p(Z_k | s_k, s_{0:k-1}^{(l)}, Z_{1:k-1}) p(s_k | s_{0:k-1}^{(l)}, Z_{1:k-1}) \\ &= \sum_{s_k} \mathcal{N}_{SE(3)}^R(Z_k(s_k); \mu_{k|k-1}^{(l)}(C), P_{k|k-1}^{(l)}(C) + Q_{k,s_k}) p(s_k) \end{aligned} \quad (\text{H.19})$$

H.1.3 Algorithme

L'algorithme permettant de réaliser l'étape de filtrage est présenté dans la table Alg.H.1. Il permet d'approcher séquentiellement la distribution $p(X_{0:k}, s_{0:k} | Z_{1:k})$ pour k allant de 1 (la première image de la vidéo) à T (la dernière image de la vidéo).

H.2 Étape de lissage

L'objectif de cette étape est d'échantillonner la distribution jointe $p(X_{1:T}, s_{1:T} | Z_{1:T})$ et nous allons voir que pour cela, le résultat de l'étape de filtrage est nécessaire ainsi que l'algorithme LG-RTS présenté dans le chapitre 3.

H.2.1 Démonstration

Nous allons détailler comment générer un échantillon $\tilde{X}_{1:T}, \tilde{s}_{1:T}$ de la distribution jointe $p(X_{1:T}, s_{1:T} | Z_{1:T})$, contrairement à l'étape de filtrage où seul les

Algorithme H.1 Étape de filtrage du LG-RBPS

- Pour $i = 1, 2, \dots, N_p$
 - Initialiser $\mu_{0|0}^{(i)}$ en choisissant aléatoirement la pose d'une des images de la base d'images avec une vitesse nulle
 - $P_{0|0}^{(i)} = P_0$ et $w_0^{(i)} = \frac{1}{N_p}$
 - FinPour
 - Pour $k = 1, 2, \dots, T$
 - Pour $i = 1, 2, \dots, N_p$
 - **Propagation du LG-EKF** : Propager $\mu_{k-1|k-1}^{(i)}$ et $P_{k-1|k-1}^{(i)}$ pour obtenir $\mu_{k|k-1}^{(i)}$ et $P_{k|k-1}^{(i)}$ en utilisant R_k et Δt (Alg.H.2)
 - **Échantillonnage optimal** : Tirer $s_k^{(i)}$ en utilisant $\mu_{k|k-1}^{(i)}$, $P_{k|k-1}^{(i)}$, Z_k et $p(s_k)$, et évaluer $p(Z_k | s_{0:k-1}^{(i)}, Z_{1:k-1})$ (Alg.H.4)
 - **Mise à jour du LG-EKF** : Mettre à jour $\mu_{k|k-1}^{(i)}$ et $P_{k|k-1}^{(i)}$ pour obtenir $\mu_{k|k}^{(i)}$ et $P_{k|k}^{(i)}$ en utilisant $y_k(s_k^{(i)})$ et Q_{CBIR} ou Q_{VM} (Alg.H.3)
 - Mise à jour du poids :
 $w_k^{(i)} = w_{k-1}^{(i)} p(Z_k | s_{0:k-1}^{(i)}, Z_{1:k-1})$
 - FinPour
 - Normaliser les poids et ré-échantillonner les particules
 - FinPour
-

Algorithme H.2 Propagation du LG-EKF

- $v = \mu_{k-1|k-1}(v)$
 - $C = \mu_{k-1|k-1}(C)$
 - $\mu_{k|k-1}(C) = \exp_{SE(3)}^{\wedge}(v\Delta t) C$
 - $\mu_{k|k-1}(v) = v$
 - $P_{k|k-1} = F P_{k-1|k-1} F^T + R$
 - où $F = \begin{bmatrix} \text{Ad}_{SE(3)}(\exp_{SE(3)}^{\wedge}(v\Delta t)) & \Phi_{SE(3)}(v\Delta t) \Delta t \\ 0 & Id \end{bmatrix}$
-

Algorithme H.3 Mise à jour du LG-EKF

- $K = P_{k|k-1} J^T \left(J P_{k|k-1} J^T + Q \right)^{-1}$
 - $\delta = K \log_{SE(3)}^{\vee} \left(Z_k \mu_{k|k-1} (C)^{-1} \right)$
 - $\mu_{k|k} (C) = \exp_{SE(3)}^{\wedge} (\delta (C)) \mu_{k|k-1} (C)$
 - $\mu_{k|k} (v) = \delta (v) + \mu_{k|k-1} (v)$
 - $P_{k|k} = \Phi_G (\delta) (Id - K J) P_{k|k-1} \Phi_G^T (\delta)$
 - où $J = \begin{bmatrix} Id & 0 \end{bmatrix}$ et $\Phi_G (\delta) = \begin{bmatrix} \Phi_{SE(3)} (\delta (C)) & 0 \\ 0 & Id \end{bmatrix}$
-

Algorithme H.4 Échantillonnage optimal

- Pour $l = 1, \dots, N$
 - $S = P_{k|k-1} (C) + Q_{VM}$
 - $e = \log_{SE(3)}^{\vee} \left(Z_k (l) \mu_{k|k-1} (C)^{-1} \right)$
 - $w (l) = \left(\sqrt{(2\pi)^6 |S|} \right)^{-1} \exp \left(-0.5 \left(e^T S^{-1} e \right) \right) p (s_k = l)$
 - FinPour
 - En pratique, pour réduire le coût calculatoire de la boucle Pour précédente, nous utilisons un arbre-kd pour retrouver les 30 plus proches voisins (parmi les composantes de Z_k) de $\mu_{k|k-1}$ et nous effectuons la boucle pour ces 30 composantes seulement.
 - Pour $l = N + 1, \dots, N + N_{CBIR}$
 - $S = P_{k|k-1} (C) + Q_{CBIR}$
 - $e = \log_{SE(3)}^{\vee} \left(Z_k (l) \mu_{k|k-1} (C)^{-1} \right)$
 - $w (l) = \left(\sqrt{(2\pi)^6 |S|} \right)^{-1} \exp \left(-0.5 \left(e^T S^{-1} e \right) \right) p (s_k = l)$
 - FinPour
 - $p \left(Z_k | s_{0:k-1}^{(i)}, Z_{1:k-1} \right) = \sum_l w (l)$
 - Faire $s_k = l$ avec probabilité $w (l)$
-

variables s_k étaient échantillonnées :

$$\begin{aligned}
& p(X_{1:T}, s_{1:T} | Z_{1:T}) \\
&= p(X_1, s_1 | X_{2:T}, s_{2:T}, Z_{1:T}) p(X_{2:T}, s_{2:T} | Z_{1:T}) \\
&= p(X_1, s_1 | X_{2:T}, s_{2:T}, Z_{1:T}) p(X_2, s_2 | X_{3:T}, s_{3:T}, Z_{1:T}) p(X_{3:T}, s_{3:T} | Z_{1:T}) \\
&= p(X_T, s_T | Z_{1:T}) \prod_{k=1}^{T-1} p(X_k, s_k | X_{k+1:T}, s_{k+1:T}, Z_{1:T}) \tag{H.20}
\end{aligned}$$

Or

$$\begin{aligned}
& p(X_k, s_k | X_{k+1:T}, s_{k+1:T}, Z_{1:T}) \\
&= \sum_{s_{1:k-1}} p(X_k, s_{1:k} | X_{k+1:T}, s_{k+1:T}, Z_{1:T}) \\
&= \sum_{s_{1:k-1}} p(X_k | X_{k+1:T}, s_{1:T}, Z_{1:T}) p(s_{1:k} | X_{k+1:T}, s_{k+1:T}, Z_{1:T}) \tag{H.21}
\end{aligned}$$

En tirant parti des propriétés de Markov de notre modèle nous avons :

$$\begin{aligned}
& p(s_{1:k} | X_{k+1:T}, s_{k+1:T}, Z_{1:T}) \\
&= p(s_{1:k} | X_{k+1}, s_{k+1}, Z_{1:k}) \\
&= \frac{p(X_{k+1}, s_{k+1} | s_{1:k}, Z_{1:k}) p(s_{1:k} | Z_{1:k})}{p(X_{k+1}, s_{k+1} | Z_{1:k})} \\
&= \frac{p(X_{k+1}, s_{k+1} | s_{1:k}, Z_{1:k}) p(s_{1:k} | Z_{1:k})}{\sum_{s_{1:k}} p(X_{k+1}, s_{1:k+1} | Z_{1:k})} \\
&= \frac{p(X_{k+1}, s_{k+1} | s_{1:k}, Z_{1:k}) p(s_{1:k} | Z_{1:k})}{\sum_{s_{1:k}} p(X_{k+1}, s_{k+1} | s_{1:k}, Z_{1:k}) p(s_{1:k} | Z_{1:k})} \\
&= \frac{p(X_{k+1} | s_{1:k+1}, Z_{1:k}) p(s_{k+1} | s_k) p(s_{1:k} | Z_{1:k})}{\sum_{s_{1:k}} p(X_{k+1} | s_{1:k+1}, Z_{1:k}) p(s_{k+1} | s_k) p(s_{1:k} | Z_{1:k})} \tag{H.22}
\end{aligned}$$

La sortie du LG-RBPF nous donne :

$$p(s_{1:k} | Z_{1:k}) \approx \sum_{l=1}^{N_p} w_k^{(l)} \delta(s_{1:k} - s_{1:k}^{(l)}) \tag{H.23}$$

En remplaçant $p(s_{1:k} | Z_{1:k})$ par (H.23) dans (H.22), nous obtenons pour le dénominateur :

$$\begin{aligned}
& \sum_{s_{1:k}} p(X_{k+1} | s_{1:k+1}, Z_{1:k}) p(s_{k+1} | s_k) \left(\sum_{l=1}^{N_p} w_k^{(l)} \delta(s_{1:k} - s_{1:k}^{(l)}) \right) \\
&= \sum_{l=1}^{N_p} w_k^{(l)} p(X_{k+1} | s_{1:k}^{(l)}, s_{k+1}, Z_{1:k}) p(s_{k+1} | s_k^{(l)}) \tag{H.24}
\end{aligned}$$

Pour le numérateur nous obtenons :

$$\begin{aligned} & p(X_{k+1}|s_{1:k+1}, Z_{1:k}) p(s_{k+1}|s_k) \left(\sum_{l=1}^{N_p} w_k^{(l)} \delta(s_{1:k} - s_{1:k}^{(l)}) \right) \\ &= \sum_{l=1}^{N_p} w_k^{(l)} p(X_{k+1}|s_{1:k}^{(l)}, s_{k+1}, Z_{1:k}) p(s_{k+1}|s_k^{(l)}) \delta(s_{1:k} - s_{1:k}^{(l)}) \end{aligned} \quad (\text{H.25})$$

Par conséquent :

$$\begin{aligned} & p(s_{1:k}|X_{k+1:T}, s_{k+1:T}, Z_{1:k}) \\ &\approx \sum_{l=1}^{N_p} \frac{w_k^{(l)} p(X_{k+1}|s_{1:k}^{(l)}, s_{k+1}, Z_{1:k}) p(s_{k+1}|s_k^{(l)})}{\sum_{m=1}^{N_p} w_k^{(m)} p(X_{k+1}|s_{1:k}^{(m)}, s_{k+1}, Z_{1:k}) p(s_{k+1}|s_k^{(m)})} \delta(s_{1:k} - s_{1:k}^{(l)}) \\ &= \sum_{l=1}^{N_p} w_{k|k+1}^{(l)} \delta(s_{1:k} - s_{1:k}^{(l)}) \end{aligned} \quad (\text{H.26})$$

où

$$\begin{aligned} w_{k|k+1}^{(l)} &= \frac{w_k^{(l)} p(X_{k+1}|s_{1:k}^{(l)}, s_{k+1}, Z_{1:k}) p(s_{k+1}|s_k^{(l)})}{\sum_{m=1}^{N_p} w_k^{(m)} p(X_{k+1}|s_{1:k}^{(m)}, s_{k+1}, Z_{1:k}) p(s_{k+1}|s_k^{(m)})} \\ &\propto w_k^{(l)} p(X_{k+1}|s_{1:k}^{(l)}, s_{k+1}, Z_{1:k}) p(s_{k+1}|s_k^{(l)}) \end{aligned} \quad (\text{H.27})$$

Et nous obtenons ainsi :

$$\begin{aligned} & p(X_k, s_k|X_{k+1:T}, s_{k+1:T}, Z_{1:T}) \\ &= \sum_{s_{1:k-1}} p(X_k|X_{k+1:T}, s_{1:T}, Z_{1:T}) p(s_{1:k}|X_{k+1:T}, s_{k+1:T}, Z_{1:k}) \\ &\approx \sum_{s_{1:k-1}} p(X_k|X_{k+1:T}, s_{1:T}, Z_{1:T}) \sum_{l=1}^{N_p} w_{k|k+1}^{(l)} \delta(s_{1:k} - s_{1:k}^{(l)}) \\ &\approx \sum_{l=1}^{N_p} w_{k|k+1}^{(l)} p(X_k|X_{k+1:T}, s_{1:k}^{(l)}, s_{k+1:T}, Z_{1:T}) \delta(s_k - s_k^{(l)}) \end{aligned} \quad (\text{H.28})$$

Finalement, sachant $\tilde{X}_{k+1:T}$ et $\tilde{s}_{k+1:T}$, \tilde{s}_k peut être échantillonné de la manière suivante ($\bar{s}_{1:k}$ est une variable temporaire) :

$$\bar{s}_{1:k} \sim \sum_{l=1}^{N_p} w_{k|k+1}^{(l)} \delta(s_{1:k} - s_{1:k}^{(l)}) \quad (\text{H.29})$$

où les poids sont calculés ainsi :

$$w_{k|k+1}^{(l)} \propto w_k^{(l)} p(\tilde{s}_{k+1}|s_k^{(l)}) \mathcal{N}_{SE(3) \times \mathbb{R}^6}(\tilde{X}_{k+1}; \mu_{k+1|k}^{(l)}, P_{k+1|k}^{(l)}) \quad (\text{H.30})$$

Et on prend :

$$\tilde{s}_k = \bar{s}_k \quad (\text{H.31})$$

Puis, \tilde{X}_k est échantillonné depuis :

$$\tilde{X}_k \sim p\left(X_k | \tilde{X}_{k+1:T}, \bar{s}_{1:k}, \tilde{s}_{k+1:T}, Z_{1:T}\right) \approx \mathcal{N}_{SE(3) \times \mathbb{R}^6}\left(X_k; \tilde{\mu}_{k|T}, \tilde{P}_{k|T}\right) \quad (\text{H.32})$$

où $\tilde{\mu}_{k|k}$ et $\tilde{P}_{k|k}$ sont obtenus en utilisant le lisseur de Rauch-Tung-Striebel sur groupe de Lie présenté dans le chapitre 3.

En itérant les quatre équations précédentes sur toute la durée de la vidéo, on peut obtenir un échantillon de $p(X_{1:T}, s_{1:T} | Z_{1:T})$. Nous allons réaliser cette opération N_p fois.

H.2.2 Algorithme

L'algorithme permettant de réaliser l'étape de lissage, c'est à dire de tirer des échantillons de la distribution jointe $p(X_{1:T}, s_{1:T} | Z_{1:T})$, est présenté en table Alg.H.5.

Algorithme H.5 LG-RBPS

- Pour $i = 1, 2, \dots, N_p$
 - Faire $j = l$ avec probabilité $w_T^{(l)}$
 - $\tilde{s}_T^{(i)} = s_T^{(j)}$, $\tilde{\mu}_{T|T}^{(i)} = \mu_{T|T}^{(j)}$, $\tilde{P}_{T|T} = P_{T|T}^{(j)}$
 - Tirer $X_T^{(i)} \sim \mathcal{N}_{SE(3) \times \mathbb{R}^6}^R(X_T; \tilde{\mu}_{T|T}^{(i)}, \tilde{P}_{T|T}^{(i)})$
 - Pour $k = T - 1, T - 2, \dots, 1$
 - Pour $l = 1, 2, \dots, N_p$
 - Donner à r la valeur de $\mathcal{N}_{SE(3) \times \mathbb{R}^6}^R(X_{k+1}; \mu_{k+1|k}^{(l)}, P_{k+1|k}^{(l)})$ évaluée en $X_{k+1} = X_{k+1}^{(i)}$
 - $w_{k|k+1}^{(l)} \propto w_k^{(l)} p(\tilde{s}_{k+1}^{(i)}) r$
 - FinPour
 - Faire $j = l$ avec probabilité $w_{k|k+1}^{(l)}$
 - $\tilde{s}_k^{(i)} = s_k^{(j)}$, $\tilde{\mu}_{k|k}^{(i)} = \mu_{k|k}^{(j)}$ et $\tilde{P}_{k|k} = P_{k|k}^{(j)}$
 - **Algorithme LG-RTS** : Lisser $\tilde{\mu}_{k|k}^{(i)}$ et $\tilde{P}_{k|k}^{(i)}$ pour obtenir $\tilde{\mu}_{k|T}^{(i)}$ et $\tilde{P}_{k|T}^{(i)}$ en utilisant $\tilde{\mu}_{k+1|T}^{(i)}$ et $\tilde{P}_{k+1|T}^{(i)}$ (Alg.[H.6](#))
 - Tirer $X_k^{(i)} \sim \mathcal{N}_{SE(3) \times \mathbb{R}^6}^R(X_k; \tilde{\mu}_{k|T}^{(i)}, \tilde{P}_{k|T}^{(i)})$
 - FinPour
 - FinPour
-

Algorithme H.6 LG-RTS

- $L = P_{k|k} F^T (R + F P_{k|k} F^T)^{-1}$ où F a été défini dans l'Alg.[H.2](#)
 - $r = \left[\begin{array}{c} \log_{SE(3)}^\vee(\mu_{k+1|T}(C) \mu_{k+1|k}^{-1}(C)) \\ \mu_{k+1|T}(v) - \mu_{k+1|k}(v) \end{array} \right]$
 - $\delta = Lr$
 - $\mu_{k|T}(C) = \exp_{SE(3)}^\wedge(\delta(1:6)) \mu_{k|k}(C)$
 - $\mu_{k|T}(v) = \delta(7:12) + \mu_{k|k}(v)$
 - $P_{k|T} = \Phi_G(\delta) (P_{k|k} + L(\varphi_G(r) P_{k+1|T} \varphi_G^T(r) - P_{k+1|k}) L^T) \Phi_G^T(\delta)$ où Φ_G a été défini dans l'Alg.[H.3](#)
-

Bibliographie

- [Abbaspour 2007] H. ABBASPOUR et M. MOSKOWITZ. *Basic Lie Theory*. World Scientific, 2007 (cf. p. 12).
- [Absil 2009] P.-A. ABSIL, R. MAHONY et R. SEPULCHRE. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009 (cf. p. 85, 89, 90).
- [Ado 1947] I. ADO. « The representation of Lie algebras by matrices ». In : *Uspekhi Matematicheskikh Nauk* 2.6 (1947), p. 159–173 (cf. p. 12).
- [Agrawal 2006] M. AGRAWAL. « A Lie Algebraic Approach for Consistent Pose Registration for General Euclidean Motion ». In : *IROS*. 2006, p. 1891–1897 (cf. p. 105, 108, 110–112).
- [Arasaratna 2010] I. ARASARATNAM, S. HAYKIN et T. HURD. « Cubature Kalman filtering for continuous-discrete systems : Theory and simulations ». In : *IEEE Transactions on Signal Processing* (2010), p. 4977–4993 (cf. p. 44).
- [Arnaudon 1992] M. ARNAUDON. « Connexions et martingales dans les groupes de Lie ». In : *Séminaire de Probabilités XXVI*. Springer, 1992, p. 146–156 (cf. p. 53).
- [Arnaudon 2014] M. ARNAUDON et L. MICLO. « A stochastic algorithm finding generalized means on compact manifolds ». In : *Stochastic Processes and their Applications* 124.10 (2014), p. 3463–3479 (cf. p. 185).
- [Arsigny 2006] V. ARSIGNY, O. COMMOWICK, X. PENNEC et N. AYACHE. « A log-euclidean framework for statistics on diffeomorphisms ». In : *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2006*. Springer, 2006, p. 924–931 (cf. p. 59).
- [Bandeira 2012] A. S. BANDEIRA, A. SINGER et D. A. SPIELMAN. « A Cheeger inequality for the graph connection Laplacian ». In : *arXiv preprint arXiv :1204.3873* (2012) (cf. p. 112).
- [Barber 2012] D. BARBER. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012 (cf. p. 88).
- [Barfoot 2014] T. D. BARFOOT et P. T. FURGALE. « Associating Uncertainty With Three-Dimensional Poses for Use in Estimation Problems ». In : *IEEE Trans. Robot.* 30.3 (juin 2014), p. 679–693. ISSN : 1941-0468 (cf. p. 30, 31, 33, 150).

- [Barrau 2014] A. BARRAU et S. BONNABEL. « Invariant Particle Filtering with application to localization ». In : *IEEE Conference on Decision and Control*. 2014 (cf. p. 173).
- [Barrau 2015] A. BARRAU et S. BONNABEL. « Intrinsic filtering on Lie groups with applications to attitude estimation ». In : *Automatic Control, IEEE Transactions on* 60.2 (2015), p. 436–449 (cf. p. 45).
- [Bay 2006] H. BAY, T. TUYTELAARS et L. VAN GOOL. « Surf : Speeded up robust features ». In : *Computer Vision–ECCV 2006*. Springer, 2006, p. 404–417 (cf. p. 140, 167).
- [BayroCorro 2000] E. BAYRO-CORROCHANO et Y. ZHANG. « The Motor Extended Kalman Filter : A Geometric Approach for Rigid Motion Estimation ». In : *Journal of Mathematical Imaging and Vision* 13.3 (2000), p. 205–228 (cf. p. 45).
- [BayroCorro 2010] E. BAYRO-CORROCHANO et G. SCHEUERMANN. *Geometric algebra computing : in engineering and computer science*. Springer Science & Business Media, 2010 (cf. p. 45).
- [Beevers 2007] K. R. BEEVERS et W. H. HUANG. « Fixed-lag sampling strategies for particle filtering SLAM ». In : *Robotics and Automation, 2007 IEEE International Conference on*. IEEE. 2007, p. 2433–2438 (cf. p. 173).
- [Bell 1993] B. M. BELL et F. W. CATHEY. « The iterated Kalman filter update as a Gauss-Newton method ». In : *Automatic Control, IEEE Transactions on* 38.2 (1993), p. 294–297 (cf. p. 85, 88, 91, 93).
- [Benhimane 2007] S. BENHIMANE et E. MALIS. « Homography-based 2d visual tracking and servoing ». In : *The International Journal of Robotics Research* 26.7 (2007), p. 661–676 (cf. p. 26).
- [Berger 2015] J. BERGER, A. NEUFELD, F. BECKER, F. LENZEN et C. SCHNÖRR. « Second Order Minimum Energy Filtering on SE3 with Nonlinear Measurement Equations ». In : *Scale Space and Variational Methods in Computer Vision : 5th International Conference, SSVM 2015, Lège-Cap Ferret, France, May 31-June 4, 2015, Proceedings*. T. 9087. Springer. 2015, p. 397 (cf. p. 173).
- [Bertsekas 1996] D. P. BERTSEKAS. « Incremental Least Squares Methods and the Extended Kalman Filter ». In : *SIAM Journal on Optimization* 6.3 (août 1996), p. 807–822 (cf. p. 85, 91).
- [Björck 1996] A. BJÖRCK. *Numerical methods for least squares problems*. Siam, 1996 (cf. p. 86).
- [Bonnabel 2007] S. BONNABEL. « Left-invariant extended Kalman filter and attitude estimation ». In : *IEEE Conference on Decision and Control*. 2007 (cf. p. 45).

- [Bonnabel 2009] S. BONNABEL, P. MARTIN et E. SALAUN. « Invariant Extended Kalman Filter : theory and application to a velocity-aided attitude estimation problem ». In : *IEEE Conference on Decision and Control and 28th Chinese Control Conference*. 2009 (cf. p. 45, 46, 49, 173).
- [Bonnabel 2013] S. BONNABEL. « Stochastic gradient descent on Riemannian manifolds ». In : *Automatic Control, IEEE Transactions on* 58.9 (2013), p. 2217–2229 (cf. p. 185).
- [Boumal 2013a] N. BOUMAL, A. SINGER, P.-A. ABSIL et V. D. BLONDEL. « Cramer-Rao bounds for synchronization of rotations ». In : *Information and Inference* (sept. 2013) (cf. p. 106).
- [Boumal 2013b] N. BOUMAL, A. SINGER et P.-A. ABSIL. « Robust estimation of rotations from relative measurements by maximum likelihood ». In : *Proceedings of the 52nd Conference on Decision and Control, CDC*. 2013 (cf. p. 105, 112).
- [Caballero 2007] F. CABALLERO, L. MERINO, J. FERRUZ et A. OLLERO. « Homography based Kalman filter for mosaic building. applications to UAV position estimation ». In : *Robotics and Automation, 2007 IEEE International Conference on*. IEEE. 2007, p. 2004–2009 (cf. p. 108).
- [Chatterjee 2013] A. CHATTERJEE et V. M. GOVINDU. « Efficient and Robust Large-Scale Rotation Averaging ». In : *ICCV*. 2013 (cf. p. 86, 112–115, 117, 118, 120).
- [Chirikjian 2012] G. S. CHIRIKJIAN. *Stochastic Models, Information Theory, and Lie Groups, Volume 2*. Springer-Verlag, 2012. ISBN : 978-0-8176-4943-2 (cf. p. 12).
- [Chirikjian 2014] G. CHIRIKJIAN et M. KOBILAROV. « Gaussian approximation of non-linear measurement models on Lie groups ». In : *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. Déc. 2014, p. 6401–6406 (cf. p. 35, 185).
- [Clemente 2007] L. A. CLEMENTE, A. J. DAVISON, I. D. REID, J. NEIRA et J. D. TARDÓS. « Mapping Large Loops with a Single Hand-Held Camera. » In : *Robotics : Science and Systems*. T. 2. 2007 (cf. p. 136, 137).
- [Conaire 2009] C. Ó. CONAIRE, M. BLIGHE et N. E. O’CONNOR. « Sensecam image localisation using hierarchical SURF trees ». In : *Advances in Multimedia Modeling*. Springer, 2009, p. 15–26 (cf. p. 163, 167).
- [Crandall 2011] D. J. CRANDALL, A. OWENS, N. SNAVELY et D. HUTTENLOCHER. « Discrete-continuous optimization for large-scale structure from motion ». In : *CVPR*. 2011, p. 3001–3008 (cf. p. 112).
- [Crassidis 2003] J. CRASSIDIS et F. MARKLEY. « Unscented Filtering for Spacecraft Attitude Estimation ». In : *Journal of Guidance, Control, and Dynamics* 26 (2003), p. 536–542 (cf. p. 45, 86).

- [Cummins 2008] M. J. CUMMINS et P. M. NEWMAN. « FAB-MAP : Probabilistic Localization and Mapping in the Space of Appearance ». In : *I. J. Robotic Res.* 27.6 (2008), p. 647–665 (cf. p. 163).
- [Davison 2007] A. J. DAVISON, I. D. REID, N. D. MOLTON et O. STASSE. *MonoSLAM : Real-time single camera SLAM*. 2007 (cf. p. 65, 86).
- [Doucet 2001] A. DOUCET, N. de FREITAS et N. GORDON. *Sequential Monte Carlo methods in practice*. Sous la dir. de SPRINGER. Springer, 2001 (cf. p. 44).
- [Dovgalecs 2011] V. DOVGALECS. « Indoor location estimation using a wearable camera with application to the monitoring of persons at home ». Thèse de doct. 2011 (cf. p. 163).
- [Eade 2007] E. EADE et T. DRUMMOND. « Monocular SLAM as a graph of coalesced observations ». In : *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE. 2007, p. 1–8 (cf. p. 136).
- [Eade 2008] E. EADE. « Monocular simultaneous localisation and mapping ». Thèse de doct. 2008 (cf. p. 136, 137, 163).
- [Endres 2012] F. ENDRES, J. HESS, N. ENGELHARD, J. STURM, D. CREMERS et W. BURGARD. « An evaluation of the RGB-D SLAM system ». In : *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE. 2012, p. 1691–1696 (cf. p. 153, 154).
- [Engel 2013] J. ENGEL, J. STURM et D. CREMERS. « Semi-Dense Visual Odometry for a Monocular Camera ». In : *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE. 2013, p. 1449–1456 (cf. p. 153, 154).
- [Engel 2014] J. ENGEL, T. SCHOPS et D. CREMERS. « LSD-SLAM : Large-Scale Direct Monocular SLAM ». In : *ECCV, Lecture Notes in Computer Science* (2014), p. 834–849. ISSN : 1611-3349 (cf. p. 137, 152–155, 159, 160, 163, 185).
- [Engels 2005] C. ENGELS, H. STEWÉNIUS et D. NISTÉR. « Bundle adjustment rules ». In : *PCV*. 2005 (cf. p. 137).
- [Enqvist 2011] O. ENQVIST, F. KAHL et C. OLSSON. « Non-sequential structure from motion ». In : *ICCV Workshops*. 2011, p. 264–271 (cf. p. 113).
- [Estrada 2005] C. ESTRADA, J. NEIRA et J. D. TARDÓS. « Hierarchical SLAM : real-time accurate mapping of large environments ». In : *Robotics, IEEE Transactions on* 21.4 (2005), p. 588–596 (cf. p. 137).
- [Faraut 2008] J. FARAUT. *Analysis on Lie groups : an introduction*. T. 110. Cambridge University Press, 2008 (cf. p. 15, 19, 33, 39).
- [Fisher 1949] R. A. FISHER, F. YATES et al. « Statistical tables for biological, agricultural and medical research. » In : *Statistical tables for biological, agricultural and medical research*. Ed. 3. (1949) (cf. p. 105, 144).

- [Fong 2002] W. FONG, S. J. GODSILL, A. DOUCET et M. WEST. « Monte Carlo smoothing with application to audio signal enhancement ». In : *Signal Processing, IEEE Transactions on* 50.2 (2002), p. 438–449 (cf. p. 174, 176, 213).
- [Fredriksso 2012] J. FREDRIKSSON et C. OLSSON. « Simultaneous Multiple Rotation Averaging using Lagrangian Duality ». In : *Asian Conference of Computer Vision*. 2012 (cf. p. 106).
- [Geiger 2013] A. GEIGER, P. LENZ, C. STILLER et R. URTASUN. « Vision meets Robotics : The KITTI Dataset ». In : *International Journal of Robotics Research (IJRR)* (2013) (cf. p. 153).
- [Gilmore 2008] R. GILMORE. *Lie Groups, Physics, and Geometry : An Introduction for Physicists, Engineers and Chemists*. Cambridge University Press, 2008 (cf. p. 6, 12).
- [González D 2013] I. GONZÁLEZ DÍAZ, V. BUSO, J. BENOIS-PINEAU, G. BOURMAUD et R. MEGRET. « Modeling instrumental activities of daily living in egocentric vision as sequences of active objects and context for alzheimer disease research ». In : *Proceedings of the 1st ACM international workshop on Multimedia indexing and information retrieval for healthcare*. ACM. 2013, p. 11–14 (cf. p. 134).
- [Govindu 2001] V. M. GOVINDU. « Combining Two-view Constraints For Motion Estimation ». In : *CVPR (2)*. 2001, p. 218–225 (cf. p. 106).
- [Govindu 2004] V. M. GOVINDU. « Lie-Algebraic Averaging for Globally Consistent Motion Estimation ». In : *CVPR (1)*. 2004, p. 684–691 (cf. p. 86, 108, 110–112).
- [Grisetti 2012] G. GRISSETTI, R. KUEMMERLE et K. NI. « Robust Optimization of Factor Graphs by Using Condensed Measurements ». In : *IEEE International Conference on Intelligent Robots and Systems (IROS)*. 2012 (cf. p. 86).
- [Hall 2003] B. HALL. *Lie Groups, Lie Algebras, and Representations An Elementary Introduction*. Sous la dir. de SPRINGER. Springer, 2003 (cf. p. 12).
- [Hall 2008] J. HALL, N. KNOEBEL et T. MCLAIN. « Quaternion Attitude Estimation for Miniature Air Vehicles Using a Multiplicative Extended Kalman ». In : *Position, Location and Navigation Symposium*. 2008, p. 1230–1237 (cf. p. 45).
- [Ham 2005] J. HAM, Y. LIN et D. D. LEE. « Learning nonlinear appearance manifolds for robot localization ». In : *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE. 2005, p. 2971–2976 (cf. p. 165).
- [Hartley 2000] R. HARTLEY et A. ZISSERMAN. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000 (cf. p. 138).
- [Hartley 2011] R. HARTLEY, K. AFTAB et J. TRUMPF. « L1 rotation averaging using the Weiszfeld algorithm ». In : *CVPR 2011*. Institute of Electrical et Electronics Engineers, juin 2011, p. 3041–3048. ISBN : 978-1-4577-0394-2 (cf. p. 112).

- [Hartley 2013] R. HARTLEY, J. TRUMPF, Y. DAI et H. LI. « Rotation Averaging ». English. In : *International Journal of Computer Vision* (2013), p. 1–39. ISSN : 0920-5691 (cf. p. 106).
- [Hauberg 2013] S. HAUBERG, F. LAUZE et K. S. PEDERSEN. « Unscented Kalman Filtering on Riemannian Manifolds ». In : *Journal of Mathematical Imaging and Vision* (2013), p. 1–18 (cf. p. 44, 46).
- [Haykin 2001] S. S. HAYKIN et al. *Kalman filtering and neural networks*. Wiley Online Library, 2001 (cf. p. 85).
- [Hervier 2012] T. HERVIER, S. BONNABEL et F. GOULETTE. « Accurate 3D maps from depth images and motion sensors via nonlinear Kalman filtering ». In : *CoRR* abs/1205.3727 (2012) (cf. p. 45).
- [Higham 2008] N. J. HIGHAM. *Functions of matrices : theory and computation*. Siam, 2008 (cf. p. 30).
- [Ireland 1990] K. IRELAND et M. I. ROSEN. *A Classical Introduction to Modern Number Theory*. T. 84. Springer Science & Business Media, 1990 (cf. p. 19).
- [Jeong 2010] Y. JEONG, D. NISTER, D. STEEDLY, R. SZELISKI et I. KWEON. « Pushing the Envelope of Modern Methods for Bundle Adjustment ». In : *CVPR*. 2010 (cf. p. 86).
- [Jiang] N. JIANG, Z. CUI et P. TAN. « A Global Linear Method for Camera Pose Registration ». In : *2013 IEEE International Conference on Computer Vision* () (cf. p. 113).
- [Johnson 1994] N. L. JOHNSON, S. KOTZ et N. BALAKRISHNAN. *Continuous Univariate Distributions*. 2nd. T. 1. Wiley, 1994 (cf. p. 39).
- [Julier 1997] S. J. JULIER et J. K. UHLMANN. « New extension of the Kalman filter to nonlinear systems ». In : *Signal Processing, Sensor Fusion, and Target Recognition VI* (juil. 1997). Sous la dir. d'I. KADAR (cf. p. 67).
- [Kaess 2008] M. KAESS, A. RANGANATHAN et F. DELLAERT. « iSAM : Incremental Smoothing and Mapping ». In : *IEEE Trans. on Robotics (TRO)* 24.6 (déc. 2008), p. 1365–1378 (cf. p. 86).
- [Kaess 2012] M. KAESS, H. JOHANSSON, R. ROBERTS, V. ILA, J. J. LEONARD et F. DELLAERT. « iSAM2 : Incremental smoothing and mapping using the Bayes tree ». In : *I. J. Robotic Res.* 31.2 (2012), p. 216–235 (cf. p. 86).
- [Kerl 2013] C. KERL, J. STURM et D. CREMERS. « Dense visual SLAM for RGB-D cameras ». In : *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE. 2013, p. 2100–2106 (cf. p. 153, 154).
- [Khurd 2012] P. KHURD, L. GRADY, R. OKETOKOUN, H. SUNDAR, T. GAJERA, S. GIBBS-STRAUSS, J. V. FRANGIONI et A. KAMEN. « Global error minimization in image mosaicing using graph connectivity and its applications in microscopy ». In : *Journal of Pathology Informatics* 2, S8 (2012) (cf. p. 108).

- [Kirillov 1994] A. KIRILLOV. *Representation Theory and Noncommutative Harmonic Analysis I : Fundamental Concepts. Representations of Virasoro and Affine Algebras*. T. 2. Springer Science & Business Media, 1994 (cf. p. 33).
- [Klein 2007] G. KLEIN et D. MURRAY. « Parallel Tracking and Mapping for Small AR Workspaces ». In : *International Symposium on Mixed and Augmented Reality (ISMAR)*. 2007 (cf. p. 86, 116, 153, 154).
- [Kolar 1993] I. KOLAR, P. MICHOR et J. SLOVAK. *Natural Operations in Differential Geometry*. Springer, 1993 (cf. p. 12).
- [Konolige 2008] K. KONOLIGE et M. AGRAWAL. « FrameSLAM : From Bundle Adjustment to Real-Time Visual Mapping ». In : *IEEE Transactions on Robotics* 24.5 (2008), p. 1066–1077 (cf. p. 135).
- [Konolige 2010] K. KONOLIGE. « Sparse Sparse Bundle Adjustment ». In : *BMVC*. 2010, p. 1–11 (cf. p. 86, 140).
- [Kraft 2003] E. KRAFT. « A quaternion-based unscented Kalman filter for orientation tracking ». In : *International Conference on Information Fusion*. T. 1. 2003, p. 47–54 (cf. p. 45).
- [Kroeger 2014] T. KROEGER et L. VAN GOOL. « Video registration to SfM models ». In : *Computer Vision–ECCV 2014*. Springer, 2014, p. 1–16 (cf. p. 163).
- [Kwon 2010] J. KWON et K. M. LEE. « Monocular SLAM with locally planar landmarks via geometric Rao-Blackwellized particle filtering on lie groups ». In : *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, p. 1522–1529 (cf. p. 173).
- [Latif 2013] Y. LATIF, C. CADENA et J. NEIRA. « Robust loop closing over time for pose graph SLAM ». In : *The International Journal of Robotics Research* (2013) (cf. p. 144).
- [Lefferts 1982] E. LEFFERTS, F. MARKLEY et M. SHUSTER. « Kalman Filtering for Spacecraft Attitude Estimation ». In : *Journal of Guidance, Control and Dynamics* (1982) (cf. p. 45, 86).
- [Li 2009] G. LI, Y. LIU, J. YIN et Z. SHI. « Newton geodesic optimization on special linear group ». In : *Proceedings of the 48th IEEE Conference on Decision and Control (CDC)* (2009) (cf. p. 86).
- [Li 2012] Y. LI, N. SNAVELY, D. HUTTENLOCHER et P. FUA. « Worldwide pose estimation using 3d point clouds ». In : *Computer Vision–ECCV 2012*. Springer, 2012, p. 15–29 (cf. p. 134).
- [Liang 2013] J. Z. LIANG, N. CORSO, E. TURNER et A. ZAKHOR. « Image based localization in indoor environments ». In : *Computing for Geospatial Research and Application (COM. Geo), 2013 Fourth International Conference on*. IEEE. 2013, p. 70–75 (cf. p. 163, 167).

- [Lim 2014] H. LIM, J. LIM et H. J. KIM. « Real-Time 6-DOF Monocular Visual SLAM in a Large-Scale Environment ». In : *ICRA*. 2014 (cf. p. 137, 152, 153, 155–157).
- [Long 2012] A. W. LONG, K. C. WOLFE, M. MASHNER et G. S. CHIRIKJIAN. « The Banana Distribution is Gaussian : A Localization Study with Exponential Coordinates. » In : *Robotics : Science and Systems*. 2012 (cf. p. 25, 30, 39).
- [Lourengo 2012] M. LOURENÇO, V. PEDRO et J. P. BARRETO. « Localization in indoor environments by querying omnidirectional visual maps using perspective images ». In : *ICRA*. 2012, p. 2189–2195 (cf. p. 163).
- [Lucas 1981] B. D. LUCAS et T. KANADE. « An iterative image registration technique with an application to stereo vision ». In : *Proceedings of the 7th international joint conference on Artificial intelligence-Volume 2*. 1981, p. 674–679 (cf. p. 138).
- [Lui 2011] Y. LUI. « Advances in matrix manifolds for computer vision ». In : *Image and Vision Computing* 30 (2011), p. 380–388 (cf. p. 85).
- [Malis 2009] E. MALIS, T. HAMEL, R. MAHONY et P. MORIN. « Dynamic estimation of homography transformations on the special linear group of visual servo control ». In : *IEEE Conference on Robotics and Automation*. 2009 (cf. p. 86).
- [Markley 2003] F. MARKLEY. « Attitude Error Representation for Kalman Filtering ». In : *Journal of Guidance, Control and Dynamics* (2003) (cf. p. 45, 86).
- [Martin 2010] P. MARTIN et E. SALAUN. « Generalized Multiplicative Extended Kalman Filter for Aided Attitude and Heading Reference System ». In : *AIAA Guidance, Navigation and Control Conference*. 2010 (cf. p. 45).
- [Martinec 2007] D. MARTINEC et T. PAJDLA. « Robust rotation and translation estimation in multiview reconstruction ». In : *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE. 2007, p. 1–8 (cf. p. 106).
- [Maybeck 1979] P. MAYBECK. *Stochastic Models, Estimation, and Control*. Academic Press, 1979 (cf. p. 44, 47, 65, 71, 78, 201).
- [Mei 2009] C. MEI, G. SIBLEY et M. CUMMINS. « A Constant-Time Efficient Stereo SLAM System. » In : *BMVC*. 2009 (cf. p. 135).
- [Meidow 2011] J. MEIDOW. « Efficient video mosaicking by multiple loop closing ». In : *Photogrammetric Image Analysis*. Springer, 2011, p. 1–12 (cf. p. 86, 105, 108).
- [Miller 1972] W. MILLER. *Symmetry Groups and Their Applications*. Academic Press, 1972 (cf. p. 19).
- [Morris 2001] D. D. MORRIS. « Gauge Freedoms and Uncertainty Modeling for 3D Computer Vision ». Thèse de doct. Citeseer, 2001 (cf. p. 141).

- [Moulon 2013] P. MOULON, P. MONASSE et R. MARLET. « Global fusion of relative motions for robust, accurate and scalable structure from motion ». In : *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE. 2013, p. 3248–3255 (cf. p. 113, 136).
- [Mourikis 2009] A. MOURIKIS, N. TRAWNY, S. ROUMELIOTIS, A. JOHNSON, A. ANSAR et L. MATTHIES. « Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing ». In : *IEEE Transactions on Robotics* 25 (2009) (cf. p. 45).
- [Newcombe 2011] R. NEWCOMBE, S. LOVEGROVE et A. DAVISON. « DTAM : Dense Tracking and Mapping in Real-Time ». In : *IEEE International Conference on Computer Vision (ICCV)*. 2011 (cf. p. 185).
- [Nielsen 2013] F. NIELSEN et R. BHATIA. *Matrix information geometry*. Springer, 2013 (cf. p. 15).
- [Nister 2004] D. NISTER. « An Efficient Solution to the Five-Point Relative Pose Problem ». In : *IEEE Trans. Pattern Anal. Mach. Intell.* 26.6 (2004), p. 756–777 (cf. p. 140).
- [Olson 2012] E. OLSON et P. AGARWAL. « Inference on networks of mixtures for robust robot mapping ». In : *RSS*. 2012 (cf. p. 144).
- [Olsson 2010] C. OLSSON, A. ERIKSSON et R. HARTLEY. « Outlier removal using duality ». In : *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, p. 1450–1457 (cf. p. 135, 137, 139, 140).
- [Olsson 2011] C. OLSSON et O. ENQVIST. « Stable structure from motion for unordered image collections ». In : *Image Analysis*. Springer, 2011, p. 524–535 (cf. p. 139).
- [Park 2008] W. PARK, Y. LIU, Y. ZHOU, M. MOSES et G. S. CHIRIKJIAN. « Kinematic state estimation and motion planning for stochastic nonholonomic systems using the exponential map ». In : *Robotica* 26.4 (2008), p. 419–434 (cf. p. 51).
- [Persson 2012] S. PERSSON et I. SHARF. « Invariant Momentum-tracking Kalman Filter for attitude estimation ». In : *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. 2012, p. 592–598 (cf. p. 45, 86).
- [Poincaré 1923] H. POINCARÉ. *La science et l'hypothèse*. E. Flammarion, 1923 (cf. p. 7).
- [Quigley 2010] M. QUIGLEY, D. STAVENS, A. COATES et S. THRUN. « Sub-meter indoor localization in unmodified environments with inexpensive sensors ». In : *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE. 2010, p. 2039–2046 (cf. p. 165).
- [Ramachandr 2000] K. RAMACHANDRA. *Kalman filtering techniques for radar tracking*. CRC Press, 2000 (cf. p. 104).

- [Rentmeeste 2010] Q. RENTMEESTERS, P. ABSIL, P. VAN DOOREN, K. GALLIVAN et A. SRIVASTAVA. « An efficient particle filtering technique on the Grassmann manifold ». In : *IEEE International Conference on Acoustics Speech and Signal Processing*. 2010 (cf. p. 44, 46).
- [Roberts 2011] R. ROBERTS, S. N. SINHA, R. SZELISKI et D. STEEDLY. « Structure from motion for scenes with large duplicate structures ». In : *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE. 2011, p. 3137–3144 (cf. p. 86, 112–115, 117–119).
- [Rudkovskii 1997] M. RUDKOVSKII. « Twisted product of Lie groups ». In : *Siberian Mathematical journal* (1997), p. 969–977 (cf. p. 29).
- [Said 2012] S. SAID et J. H. MANTON. « Extrinsic Mean of Brownian Distributions on Compact Lie Groups ». In : *IEEE Transactions on Information Theory* 58.6 (2012), p. 3521–3535 (cf. p. 49).
- [Sarkka 2007] S. SARKKA. « On unscented Kalman filtering for state estimation of continuous-time nonlinear systems ». In : *Automatic Control, IEEE Transactions on* 52.9 (2007), p. 1631–1641 (cf. p. 44).
- [Särkkä 2007] S. SÄRKKÄ, A. VEHTARI et J. LAMPINEN. « Rao-Blackwellized particle filter for multiple target tracking ». In : *Information Fusion* 8.1 (2007), p. 2–15 (cf. p. 178).
- [Särkkä 2013] S. SÄRKKÄ. *Bayesian filtering and smoothing*. 3. Cambridge University Press, 2013 (cf. p. 120, 121).
- [Sattler 2011] T. SATTLER, B. LEIBE et L. KOBELT. « Fast image-based localization using direct 2D-to-3D matching ». In : *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE. 2011, p. 667–674 (cf. p. 134, 163).
- [Sattler 2012] T. SATTLER, T. WEYAND, B. LEIBE et L. KOBELT. « Image Retrieval for Image-Based Localization Revisited ». In : *BMVC*. 2012 (cf. p. 163).
- [Selig 2005] J. M. SELIG. « Lie groups and Lie algebras in robotics ». In : *NATO Science Series II : Mathematics, Physics and Chemistry* (2005), p. 101–125 (cf. p. 24, 25, 30).
- [Simon 2010] D. SIMON. « Kalman filtering with state constraints : a survey of linear and nonlinear algorithms ». In : *IET Control Theory & Applications* 4.8 (2010), p. 1303–1318 (cf. p. 44, 65, 66, 79).
- [Singer 2006] H. SINGER. « Continuous-Discrete Unscented Kalman Filtering ». In : *Technical Report* (2006) (cf. p. 67).
- [Singer 2011] A. SINGER et Y. SHKOLNISKY. « Three-dimensional structure determination from common lines in cryo-EM by eigenvectors and semidefinite programming ». In : *SIAM journal on imaging sciences* 4.2 (2011), p. 543–572 (cf. p. 112).
- [Sipos 2008] B. SIPOS. « Application of the Manifold-Constrained Unscented Kalman Filter ». In : *Position, Location and Navigation Symposium*. 2008, p. 30–43 (cf. p. 45).

- [Sivic 2009] J. SIVIC et A. ZISSERMAN. « Efficient visual search of videos cast as text retrieval ». In : *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31.4 (2009), p. 591–606 (cf. p. 142).
- [Smith 2003] P. SMITH, T. DRUMMOND et K. ROUSSOPOULOS. « Computing MAP trajectories by representing, propagating and combining PDFs over groups ». In : *ICCV*. 2003, p. 1275–1282 (cf. p. 31).
- [Snoussi 2006] H. SNOUSSI et A. MOHAMMAD-DJAFARI. « Particle Filtering on Riemannian Manifolds ». In : *AIP Conference , Vol. Issue 1, p219*. T. 872. 2006 (cf. p. 44, 46).
- [Strasdat 2010a] H. STRASDAT, J. MONTIEL et A. DAVISON. « Scale Drift-Aware Large Scale Monocular SLAM ». In : *RSS*. 2010 (cf. p. 137).
- [Strasdat 2010b] H. STRASDAT, J. M. M. MONTIEL et A. J. DAVISON. « Real-time monocular SLAM : Why filter ? » In : *ICRA*. 2010, p. 2657–2664 (cf. p. 86).
- [Strasdat 2011] H. STRASDAT, A. DAVISON, J. MONTIEL et K. KONOLIGE. « Double window optimisation for constant time visual SLAM ». In : *IEEE International Conference on Computer Vision (ICCV)*. 2011 (cf. p. 86).
- [Strasdat 2012] H. STRASDAT. « Local Accuracy and Global Consistency for Efficient Visual SLAM ». Thèse de doct. 2012 (cf. p. 26, 30, 136).
- [Sturm 2012] J. STURM, N. ENGELHARD, F. ENDRES, W. BURGARD et D. CREMERS. « A benchmark for the evaluation of RGB-D SLAM systems ». In : *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE. 2012, p. 573–580 (cf. p. 153, 154).
- [Sunderhauf 2012] N. SUNDERHAUF et P. PROTZEL. « Switchable Constraints for Robust Pose Graph SLAM ». In : *IROS*. 2012 (cf. p. 144).
- [Tan 2013] W. TAN, H. LIU, Z. DONG, G. ZHANG et H. BAO. « Robust monocular SLAM in dynamic environments ». In : *Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on*. IEEE. 2013, p. 209–218 (cf. p. 186).
- [Taylor 1994] C. J. TAYLOR et D. J. KRIEGMAN. *Minimization on the Lie group $SO(3)$ and related manifolds*. Rapp. tech. Yale University, 1994 (cf. p. 86).
- [Thrun 2005] S. THRUN, W. BURGARD et D. FOX. *Probabilistic robotics*. MIT press, 2005 (cf. p. 40).
- [Tompkins 2007] F. TOMPKINS et P. J. WOLFE. « Bayesian Filtering on the Stiefel Manifold ». In : *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*. 2007 (cf. p. 44, 46).

- [Torralba 2008] A. TORRALBA, R. FERGUS et W. T. FREEMAN. « 80 million tiny images : A large data set for nonparametric object and scene recognition ». In : *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 30.11 (2008), p. 1958–1970 (cf. p. 167).
- [Trawny 2005] N. TRAWNY et S. ROUMELIOTIS. *Indirect Kalman Filter for 3D Attitude Estimation*. Rapp. tech. University of Minnesota, 2005 (cf. p. 45).
- [Trawny 2007] N. TRAWNY, A. MOURIKIS, S. ROUMELIOTIS, A. JOHNSON et J. MONTGOMERY. « Vision-Aided Inertial Navigation for Pin-Point Landing using Observations of Mapped Landmarks ». In : *Journal of Field Robotics* 24 (2007), p. 357–378 (cf. p. 45).
- [Umeyama 1991] S. UMEYAMA. « Least-squares estimation of transformation parameters between two point patterns ». In : *IEEE Transactions on pattern analysis and machine intelligence* 13.4 (1991), p. 376–380 (cf. p. 142).
- [Wachinger 2008] C. WACHINGER, W. WEIN et N. NAVAB. « Registration strategies and similarity measures for three-dimensional ultrasound mosaicing ». In : *Academic radiology* 15.11 (2008), p. 1404–1415 (cf. p. 105).
- [Wang 2006] Y. WANG et G. CHIRIKJIAN. « Error Propagation on the Euclidean Group With Applications to Manipulators Kinematics ». In : *IEEE Transactions on Robotics* 22 (2006) (cf. p. 35).
- [Wang 2008] Y. WANG et G. S. CHIRIKJIAN. « Nonparametric Second-Order Theory of Error Propagation on Motion Groups ». In : *International Journal of Robotics Research* 27(11-12) (2008), p. 1258–1273 (cf. p. 51).
- [Wang 2013] L. WANG et A. SINGER. « Exact and stable recovery of rotations for robust synchronization ». In : *Information and Inference* 2.2 (oct. 2013), p. 145–193 (cf. p. 112).
- [Wolf 2005] J. WOLF, W. BURGARD et H. BURKHARDT. « Robust vision-based localization by combining an image-retrieval system with Monte Carlo localization ». In : *Robotics, IEEE Transactions on* 21.2 (2005), p. 208–216 (cf. p. 165).
- [Wolfe 2011] K. WOLFE, M. MASHNER et G. CHIRIKJIAN. « Bayesian Fusion on Lie Groups ». In : *Journal of Algebraic Statistics* 2 (2011), p. 75–97 (cf. p. 35).
- [Zach 2010] C. ZACH, M. KLOPSCHITZ et M. POLLEFEYS. « Disambiguating visual relations using loop constraints ». In : *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, p. 1426–1433 (cf. p. 144).
- [Zhang 2015] J. ZHANG et S. SINGH. « Visual-lidar Odometry and Mapping : Low-drift, Robust, and Fast ». In : *Submitted to IEEE International Conference on Robotics and Automation(ICRA)*. Seattle, WA, mai 2015 (cf. p. 155, 157, 158).